

2020

# Automotive Cognitive Access: Towards customized vehicular communication system

Dawood, Muhammad Ahmad

<http://hdl.handle.net/10026.1/16532>

---

<http://dx.doi.org/10.24382/1018>

University of Plymouth

---

*All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.*



# **UNIVERSITY OF PLYMOUTH**

## **Automotive Cognitive Access: Towards customized vehicular communication system**

By

**Muhammad Dawood**

A thesis submitted to the University of Plymouth in partial fulfilment for  
the degree of

**DOCTOR OF PHILOSOPHY**

School of Engineering, Computing and Mathematics

**September 2020**



## **Copyright Statement**

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.



# **Automotive Cognitive Access: Towards customized vehicular communication system**

Muhammad Dawood

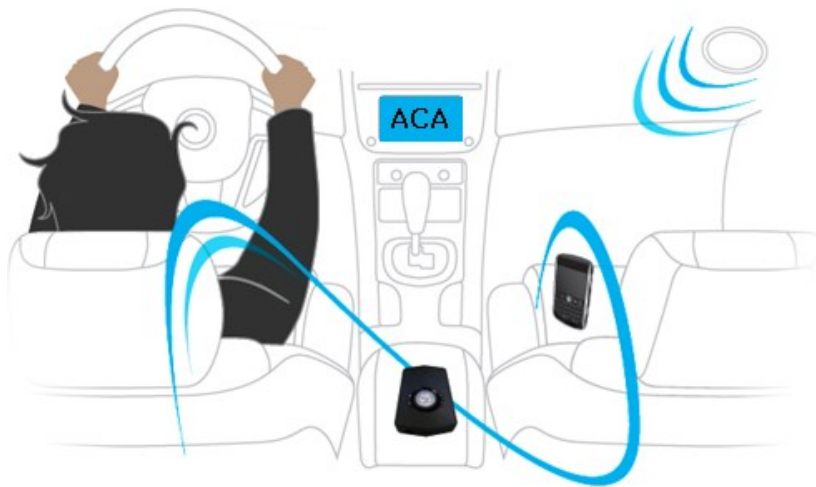
## **Abstract**

The evolution of Software Defined Networking (SDN) and Virtualization of mobile Network Functions (NFV) have enabled the new ways of managing mobile access systems and are seen as a major technological foundation of the Fifth Generation (5G) of mobile networks. With the appearance of 5G specifications, the mobile system architecture has the transition from a network of entities to a network of functions. This paradigm shift led to new possibilities and challenges. Existing mobile communication systems rely on closed and inflexible hardware-based architectures both at the access and core network. It implies significant challenges in implementing new techniques to maximize the network capacity, scalability and increasing performance for diverse data services.

This work focuses preliminary on the architectural evolutions needed to solve challenges perceived for the next generation of mobile networks. I consider Software defined plus Virtualization featured Mobile Network (S+ MN) architecture as a baseline reference model, aiming at the further improvements to support the access requirements for diverse user groups. I consider an important class of things, vehicles, which needs efficient mobile internet access at both the system and application levels. I identify and describe key requirements of emerging vehicular communications and assess existing standards to determine their limitations. To provide optimized wireless communications for the specific user group, the 5G systems come up with network slicing as a potential solution to create customized networks. Network slicing has the capability to facilitates dynamic and efficient allocation of network resources and support diverse service scenarios and services. A network slice can be broadly defined as an end-to-end logically isolated network that includes end devices as well as access and core network functions. To this effect, I describe the enhanced behaviour of S+ MN architecture for the collection of network resources and details the potential functional grouping provided by S+ MN architecture that paves the way to support automotive slicing. The proposed enhancements support seamless connection mobility addressing the automotive access use case highly mobile environment. I follow the

distribution of gateway functions to solve the problem of unnecessary long routes and delays. Exploiting the open SDN capabilities, the proposed S+ NC is able to parallelize the execution of certain control plane messages thus enabling the signalling optimisation. Furthermore, it enables the (Re)selection of efficient data plane paths with implied upper-layer service continuity mechanisms that remove the chains of IP address preservation for session continuity during IP anchor relocation.

An implementation setup validates the proposed evolutions, including its core functionalities implemented using the ns-3 network simulator. The proposed slicing scheme has been evaluated through a number of scenarios such as numbers of signalling messages processed by control entities for an intersystem handover procedure relative to current mobile network architecture. I also perform the performance improvement analysis based on simulation results. Furthermore, I experimentally prove the feasibility of using Multipath TCP for connection mobility in intersystem handover scenario. The experiments run over the Linux Kernel implementation of Multipath TCP developed over the last years. I extend the Multipath TCP path management to delegates the management of the data paths according to the application needs. The implementation results have shown that the proposed S+ MN slicing architecture and enhancements achieve benefits in multiple areas, for example improving the mobility control and management, maintaining QoS, smooth handover, session continuity and efficient slice management and orchestration.



# Contents

Abstract .....	i
Contents .....	iii
List of Figures .....	vi
List of Tables .....	viii
List of Abbreviations and Acronyms .....	ix
Acknowledgement .....	xvii
Author's Declaration .....	xviii
1 Introduction .....	20
1.1 Aims and objectives of the research .....	24
1.2 Thesis structure .....	25
2 Terminology .....	27
3 Background and State of the Art .....	33
3.1 Evolved Packet System (EPS) .....	33
3.2 State of the Art .....	37
3.3 5G System Architecture .....	37
3.4 Protocol Stack in 5G .....	41
3.4.1 Control Plane Protocol Stack .....	41
3.4.2 User Plane Protocol Stack .....	44
3.5 Mobility Enhancements in 5G System .....	45
3.6 Quality of Service (QoS) Provisioning .....	50
3.7 Evaluation and Open challenges for 5G Network .....	55
4 Design Selection .....	62
4.1 Automotive Access .....	62
4.2 Vehicular Communication use cases .....	64
4.3 Vehicular Communications Requirements .....	65
4.4 Heterogeneous environment .....	69
4.5 Intersystem internetworking .....	70
4.5.1 Access Network Discovery and Selection Function (ANDSF) ..	72



4.5.2	Media Independent Handover -----	72
4.6	Distributed Mobility Management (DMM) -----	75
4.7	Multiple PDN support-----	78
4.7.1	IPv6 multi-homing -----	79
4.7.2	Multipath TCP (MPTCP) -----	80
4.8	Important performance metrics -----	82
4.9	Key enabling technologies -----	84
4.9.1	Software Defined Networking (SDN)-----	84
4.9.2	Virtualization -----	85
4.9.3	Mobile Network Slicing -----	87
4.10	Final remarks. The need for customised vehicular access -----	89
5	S Plus Mobile Network -----	91
5.1	S+ MN Architecture Overview -----	91
5.2	S+ MN Slicing scheme -----	93
5.3	Automotive Slicing -----	97
5.4	Automotive Slice Instance -----	99
5.5	Automotive Network Slices -----	102
5.6	Exemplary Procedures -----	103
5.6.1	Initial Attachment and V2X AS Connection Setup-----	104
5.6.2	Multiple connections setup to multiple Network Slices-----	105
5.6.3	QoS provisioning procedures -----	108
5.6.4	QoS aware slice management and orchestration -----	111
5.6.5	Mobility Management procedures in automotive slicing -----	113
5.6.6	Inter RAN Handover with DPF relocation -----	114
5.6.7	Inter slice Handover with DPF relocation -----	120
6	Evaluation-----	122
6.1	Simulation Phase 1: Functional Validation setup-----	122
6.2	Realization of S+ MN Architecture -----	125
6.2.1	System Procedures-----	129
6.2.2	Initial attachment procedure -----	130

6.2.3	Intersystem handover -----	131
6.2.4	Qualitative analysis -----	132
6.2.5	Simulation Phase 1 Results and Analysis -----	133
6.3	Simulation Phase 2: QoS-aware connection mobility with Multipath TCP	137
6.3.1	Management of subflows -----	138
6.3.2	Realization of S+ Path Manager -----	140
6.3.3	Performance evaluation -----	143
7	Conclusions -----	149
7.1	Achievements of the Research -----	151
7.2	Limitations of the research -----	152
7.3	Suggested scope for future work -----	154
	References -----	155
	Appendix -----	160
	S+ Path Management Code -----	160
	S Plus Plugin -----	160
	Socket Raper -----	172
	Gateway manager python script -----	183

## List of Figures

Figure 1.1: Global Mobile Devices and Connections Growth [1].....	20
Figure 3.1: EPS architecture overview .....	33
Figure 3.2: The generalised design of the 5G System Architecture [12] .	39
Figure 3.3: Control Plane protocols between a 5G-AN and AMF over reference point N2 .....	42
Figure 3.4: Control Plane protocol stack between a 5G-AN and a SMF [12] .....	43
Figure 3.5: Control Plane between a UE and a SMF .....	44
Figure 3.6: User plane end-to-end protocol stack [42].....	45
Figure 3.7: 5G access and mobility framework .....	46
Figure 3.8: Non-roaming architecture for internetworking between 5GS and EPS.....	49
Figure 3.9: 5G QoS functional split, classification, marking and mapping to AN resource .....	52
Figure 4.1: Conceptual view of future automotive environment .....	62
Figure 4.2: V2N context applications .....	63
Figure 4.3: Heterogeneous environment .....	69
Figure 4.4: EPS architecture inter system internetworking .....	74
Figure 4.5: Baseline architecture for distributed mobility .....	76
Fig. 4.6a: Client-based approach .....	78
Figure 4.6b: Network-based approach .....	78
Figure 4.7: IPv6 multi-homing: overview and operations [12] .....	79
Figure 4.8: Multipath TCP multipath operations .....	81
Figure 4.9: Typical Open SDN Architecture.....	85
Figure 5.1: Generic view of the S+ MN architecture .....	91
Figure 5.2: S+ MN Slicing Architecture .....	95
Figure 5.3: Automotive Slice (potential) functional grouping .....	100
Figure 5.4: Overview of the proposed slice instances for the identified V2X service categories .....	103
Figure 5.5: Signalling diagram of initial attachment and service setup .	104
Figure 5.6: Service connection setup via multiple slices .....	106
Figure 5.7: Signalling diagram for service connection setup via multiple slices.....	107
Figure 5.8: QoS provisioning signaling.....	109
Figure 5.9: QoS aware slice management and orchestration .....	112
Figure 5.10: Inter RAN Handover with DPF relocation.....	115

Figure 5.11: IPv6 multi-homing only seamless connection mobility .....	118
Figure 5.12: MPTCP based operation for connection mobility .....	119
Figure 5.13: Inter Slice Handover .....	121
Figure 6.1: The OFSwitch13 module overview [25] .....	124
Figure 6.2: Control plane instantiation within S+ MN architecture. ....	126
Figure 6.3 : The OFSwitch13 module overview [25] .....	127
Figure 6.4: Class diagram of the OFSwitch13 module enhancements for multiple DPEs .....	128
Figure 6.5: Network topology of the evaluation scenario with multiple DPEs. ....	129
Figure 6.6: Initial attachment operation.....	130
Figure 6.7: Inter-System Handover operation .....	132
Figure 6.8: Intersystem handover measurements .....	134
Figure 6.9: Empirical CFD of handover measurement for ping .....	135
Figure 6.10: The packet scheduler distributes the segments on different TCP subflows .....	138
Figure 6.11: The S+ Path management with Netlink support [93].....	140
Figure 6.12: The network topology used for simulation.....	144
Figure 6.13: Initial access interface (S+ vAP) link quality for handover forecast .....	145
Figure 6.14: Active interfaces for Full-MPTCP Mode .....	146
Figure 6.15: Active interfaces for the Single-Path handover at second 8 from S+ AP to S+ AN.....	146
Figure 6.16: smooth handover using the S+ AP link quality for handover forecast. ....	147
Figure 6.17: Comparison of Kernel path manager and S+ PM.....	148

## List of Tables

Table 1: standardized 5QI to QoS characteristics mapping (adopted from [12]) .....	50
Table 2: ITS applications and their preferred communication specifications .....	68
Table 3: Defined Link delay values .....	136
Table 4: Performance improvement analysis for an Intersystem Handover scenario .....	137
Table 5: S+ path manger dependency .....	141
Table 6: Test Parameters .....	143

## List of Abbreviations and Acronyms

AAA	Authentication, Authorisation and Accounting
ACA	Automotive Cognitive Access
ACK	Acknowledgement
AF	Application Function
AMBR	Aggregate Maximum Bit Rate
AMF	Access and Mobility Management Function
AN	Access Network
ANDSF	Access Network Discovery and Selection Function
AP	Access Point
API	Application Programming Interface
APN	Access Point Name
AR	Augment Reality
ARP	Allocation and Retention Priority
ASI	Automotive Slice Instance
AUC	Authentication Centre
AUSF	Authentication Server Function
AUT	Automated
BCE	Binding Cache Entry
BER	Bit Error Rate
BR	Branching Point
BSF	Binding Support Function
BSMS	Basis Safety Messages
BTS	Base Transceiver Station
CAN	Controller Area Network
CAPIF	Common API Framework for 3GPP northbound APIs
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
CHF	Charging Function
CM	Connection Manager/Management
CoA	Care of Address
CoS	Class of Service
CP	Control Plane
CPF	Control Plane Function
CP-OFDM	Cyclic-Prefix OFDM
C-RAN	Cloud Radio Access Network

CRUD	Create, Read, Update, Delete
CSG	Closed Subscriber Group
CSMA	Carrier Sense Multiple Access
DB	Database
DÉCOR	Dedicated Core Network
DENMS	Decentralized Environmental. Notification Messages
DFT	Discrete Fourier Transform
DGI	Data Gathering Interfaces
DL	Downlink
DMM	Distributed Mobility Management
DN	Data Network
DNAI	DN Access Identifier
DNN	Data Network Name
DPE	Data Plane Entity
DRB	Data Radio Bearers
DSMIPv6	Dual Stack Mobile IPv6
DSS	Data Sequence Signal
E2E	End-to-End
EBI	EPS Bearer Identity
ECMP	Equal Cost Multipath
eHRPD	evolved High Rate Packet Data
eMBB	enhanced Mobile Broadband
eNodeB	Evolved Node B
EPC	Evolved Packet Core
ePDG	evolved Packet Data Gateway
EPS	Evolved Packet System
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved Universal Terrestrial Access Network
eV2X	enhancements V2X
FAR	Forwarding Action Rule
FCD	Floating Car Data
FE	Front End
FPE	Flow Priority Indicator
FQDN	Fully Qualified Domain Name
GBR	Guaranteed Bit Rate
GERAN	GSM EDGE Radio Access Networks
GFBR	Guaranteed Flow Bit Rate
GMLC	Gateway Mobile Location Centre

gNB	Next generation NodeB
GPRS	General Packet Radio Service
GPSI	Generic Public Subscription Identifier
GTP	GPRS Tunneling Protocol
GTP-U	GPRS Tunnelling Protocol User
GUAMI	Globally Unique AMF Identifier
HA	Home Agent
HD	High Definition
HoS	Home Address
HR	Home Routed (roaming)
H-RAN	Heterogenous RAN
HSS	Home Subscriber Server
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IMT	International Mobile Telecommunications
ITS	Intelligent Transportation System
IaaS	Infrastructure as a Service
ITU	International Telecommunication Union
JSON	Java Script Object Notation
KPI	Key Performance Indicator
LADN	Local Area Data Network
LBO	Local Break Out (roaming)
LCS	Location Services
LMA	Local Mobility Anchor
LMF	Location Management Function
LPP	LTE Positioning Protocol
LRF	Location Retrieval Function
LTE	Long Term Evolution
MAC	Medium Access Control
MAG	Mobile Access Gateway
MANO	Management and Orchestration
MC	Multi Career
MCX	Mission Critical Service
MDBV	Maximum Data Burst Volume
MEC	Mobile Edge Computing
MF	Mobility anchor Function
MFB	Maximum Flow Bitrate



MICO	Mobile Initiated Connection Only
MIMO	Multiple-Input Multiple-Output
MIPv6	Mobile IPv6
MME	Mobility Management Entity
mMTC	massive Machine Type Communications
MPS	Multimedia Priority Service
MPTCP	Multipath Transmission Control Protocol
MTCS	Multipath Transmission Control Scheme
MU	Mobile Unit
MVNO	Mobile Virtual Network Operator
N3IWF	Non-3GPP InterWorking Function
NACF	Network Access Control Function
NAI	Network Access Identifier
NAS	Non-Access Stratum
NAS-MM	Non-Access Stratum Mobility Management
NAS-SM	Non-Access Stratum Session Management
NEF	Network Exposure Function
NF	Network Function
NFV	Network Function Virtualization
NFVI	NFV Infrastructure
NGAP	Next Generation Application Protocol
Non-GBR	Non-Guaranteed Bit Rate
NR	New Radio
NRF	Network Repository Function
NS	Network Slice
NS-3	Network Simulator 3
NSI	Network Slice Instance
NSI ID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
NSSP	Network Slice Selection Policy
NSTI	Network Slice Type Indicator
NWDAF	Network Data Analytics Function
OFDM	Orthogonal Frequency-Division Multiplexing
OFS	Open Flow Switch
ONF	Open Networking Foundation
OTA	Over The Air
PAPR	Peak to Average Power Ratio

PCAP	Packet Capture
PCEF	Policy and Charging Enforcement Function
PCF	Policy Control Function
PCRF	Policy and Charging Rules Function
PDB	Packet Delay Budget
PDCCP	Packet Data Convergence Protocol
PDN	Packet Data Network
P-GW	Packet Data Network Gateway
PDP	Packet Data Protocol
PDR	Packet Detection Rule
PEI	Permanent Equipment Identifier
PER	Packet Error Rate
PFD	Packet Flow Description
PGW-C	Packet Data Network Gateway Control
PGW-U	Packet Data Network Gateway User
P-GW	PDN Gateway
PHY	Physical layer
PL	Priority Level
PoP	Point of Presence
PPD	Paging Policy Differentiation
PPF	Paging Proceed Flag
PPI	Paging Policy Indicator
PSA	PDU Session Anchor
QCI	QoS Class Identifier
QFI	QoS Flow Identifier
QKD	Quantum Key Distribution
QoE	Quality of Experience
QoS	Quality of Service
RAB	Radio Access Bearer
RAN	Radio Access Network
RAT	Radio Access Technology
RAU	Routing Area Update
RB	Radio Bearer
RLC	Radio Link Control
RM	Registration Management
RQA	Reflective QoS Attribute
RQI	Reflective QoS Indication
RRC	Radio Resource Control

RRM	Radio Resource Management
RSS	Received Signal Strength
RSU	Road Side Units
RTT	Round Trip Time
S+ MN	S+ Mobile Network
S+ AN	S+ Access Network
S+ AU	S+ Access Unit
S+ CN	S+ Core Network
S+ DS	S+ Data Server
S+ NC	S+ Network Controller
S+	SDN Plus Virtualization
SA	NR Standalone New Radio
SAE	System Architecture Evolution
SBA	Service Based Architecture
SBI	Service Based Interface
SC	Spectrum Coordinator
SCA	Slice Configuration and Allocation
SCM	Slice Coordination and Mobility
SCTP	Stream Control Transmission Protocol
SD	Slice Differentiator
SDAP	Service Data Adaptation Protocol
SDF	Service Data Flow
SDN	Software Defined Networking
SEAF	Security Anchor Functionality
SEPP	Security Edge Protection Proxy
S-GW	Serving Gateway
SIL	Slice Instance Layer
SLA	Service Level Agreement
SMF	Session Management Function
SMSF	Short Message Service Function
SN	Sequence Number
S-NSSAI	Single Network Slice Selection Assistance Information
SOL	Slice Orchestration Layer
SPR	Subscription Profile Repository
SQP	Slice QoS Provisioning
SSC	Session and Service Continuity
SSCMSP	Session and Service Continuity Mode Selection Policy
SST	Slice/Service Type

SUCI	Subscription Concealed Identifier
SUG	Slice User Group
SUPI	Subscription Permanent Identifier
TAI	Tracking Area Identity
TALs	Tracking Area Lists
TAs	Tracking Areas
TNL	Transport Network Layer
TNLA	Transport Network Layer Association
TSP	Traffic Steering Policy
TVWS	TV White Space
UDM	Unified Data Management
UDR	Unified Data Repository
UDSF	Unstructured Data Storage Function
UE	User Equipment
UHD	Ultra-High Definition
UL	Uplink
UP	User Plane
UP CL	Uplink classifier
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communications
URSP	UE Route Selection Policy
UTRAN	UMTS Terrestrial Radio Access Network
V2I	Vehicle to Infrastructure
V2X	Vehicle to Everything
V2X AS	V2X Application Service
V2V	Vehicle to Vehicle
vCPF	Virtualised Core Plane Function
vDPF	Virtual Data Plane Function
VID	VLAN Identifier
VLAN	Virtual Local Area Network
vNF	Virtualised Network Function
vPF	Virtual Policy Control Function
VR	Virtual Reality
VRU	Vulnerable Road User
WAPS	Wlan Access Points
WAVE	Wireless Access in Vehicular Environments
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

WSD	White Space Device
WSDB	White Space Database
(R)AN	(Radio) Access Network
3G	Third Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
5G-AN	5G Access Network
5GC	5G Core Network
5G-EIR	5G-Equipment Identity Register
5G-GUTI	5G Globally Unique Temporary Identifier
5GS	5G System
5QI	5G QoS Identifier

## **Acknowledgement**

I must say thanks to GOD for giving me a talent that few people possess and the means by which to use that talent.

I want to thank my supervisory team for their support. Especially I want to thank my Director of Studies, Prof. Dr Woldemar Fuhrmann, for the support and the discussions which led to many new ideas. Your advice and guidance on both research as well as on my further way to go have been invaluable for me. Thank you for being my mentor and that I had the chance to learn so much from you.

I would also like to thank my supervisor Dr Bogdan V. Ghita for your constructive words, when I needed them and for your advice, guidance, and your valuable feedback during the work.

And then there's Mr S. Shafi (Burraq Technologies Germany). Who put lots of hours into ensuring the technical guidance and took the time to talk to me, to answer my sometimes dumb questions his suggestions (and saves) have kept my facts true. Thanks, Shafi.

Thanks, too, to my dear family and friends. You are my support system. And you are there when I need you; you leave when I need space. Without your prayers, I would not be able to function nearly as effectively. Thanks, guys!

## Author's Declaration

At no time during the registration for the degree of *Doctor of Philosophy* has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

Relevant seminars and conferences were regularly attended at which work was often presented and several papers were published in the course of this research project.

### *Publications:*

M. Dawood, W. Fuhrmann, Bogdan V. Ghita, "Prospects for TV White Space access in the Evolved Packet Core environment", Proceedings of the 6th International Conference on Internet Technologies and Applications (ITA 15), 8-11 September 2015, Glyndwr University, Wrexham, UK, ISBN: 978-1-4799-8036-9, pp. 47-52, DOI: 10.1109/ITechA.2015.7317368  
DOI: <https://doi.org/10.1109/ITechA.2015.7317368>

M. Dawood, W. Fuhrmann, and B. V. Ghita, "Assay of IP mobility management in SDN based mobile network architecture", Proceedings of the 7th International Conference on Internet Technologies and Applications (ITA 17), 12-15 Sept. 2017, Glyndwr University, Wrexham, UK, ISBN: 978-1-5090-4815-1, pp. 69-74.  
DOI: <http://dx.doi.org/10.1109/ITECHA.2017.8101913>  
PEARL (OA): <http://hdl.handle.net/10026.1/13874>


M. Dawood, W. Fuhrmann, and B. V. Ghita, "Improvements for DMM in SDN and Virtualization-Based Mobile Network Architecture", International Journal of Mobile Network Communications & Telematics (IJMNCT), August 2018, pp. , volume 8, pp. 01-14.  
DOI: <http://dx.doi.org/10.5121/ijmnct.2018.8401>  
PEARL (OA): <http://hdl.handle.net/10026.1/13675>

*Presentations at conferences:*

Assay of White Space Technology Standards for Vehicular Cognitive Access, Proceedings of the Tenth International Network Conference (INC 2014), Plymouth, UK, July 8-10, ISBN: 9781841023731 pp23-33.

Assay of Multipath TCP for Session Continuity in Distributed Mobility Management, Proceedings of the Eleventh International Network Conference (INC 2016), Frankfurt, Germany, July 19-21, 2016 ISBN: 978-1-84102-410-3, pp 141-146

Word count of the main body of thesis: 45425 words.

Signed  \_\_\_\_\_

Date 25<sup>th</sup> September, 2020



## 1 Introduction

The rapid growth of global mobile devices, connections and mobile data traffic, has been widely recognized and reported [1] and the mobile communications industry is preparing to cope with a 1000x increase of traffic by 2021 over 2010 [2]. More and more people see their handheld devices as an annex of their workspace while on move (Figure 1.1) and the continuous improvement to the mobile network architecture is becoming increasingly important to support the performance requirements for the ubiquitous wireless connections.

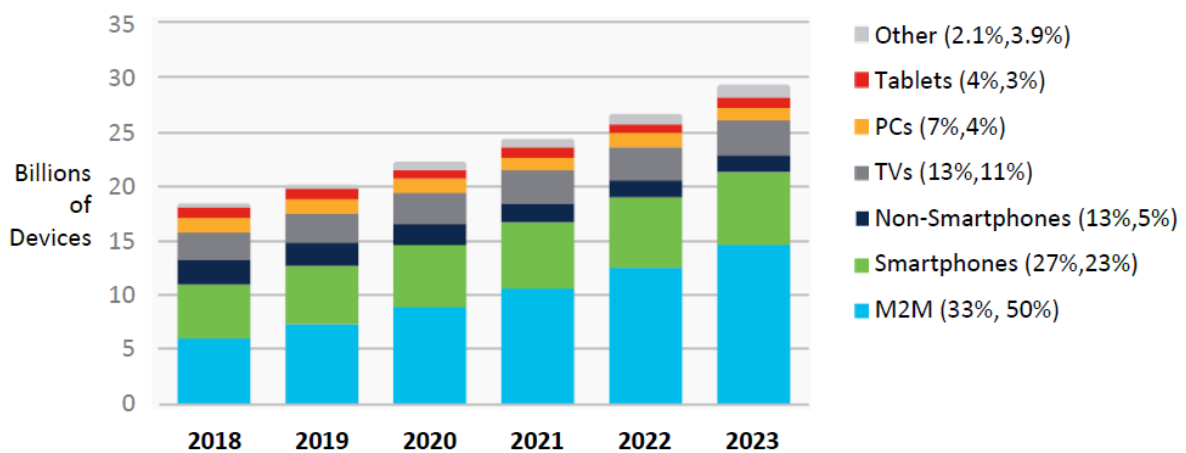


Figure 1.1: Global Mobile Devices and Connections Growth [1]

In addition, in dense traffic areas, the heavy load generated by message transmissions from several wireless enabled devices and services strongly challenge radio systems capacity and potentially penalizes the delivery of traditional applications. Due to the growing demand for wireless communications for a wide range of purposes, current radio systems may come to practical limits of frequency spectrum bands. To overcome bandwidth scarcity issue global interest in the new solution is being fuelled. Use of vacant frequencies (termed as white spaces) at a given time in a given geographical area not being used by licensed services has been identified as an important spectrum resource. The unused spectrum in UHF TV [3] broadcast band (470 -790 MHz) is one of these vacant frequencies and is referred as TV white space [4] providing excellent propagation characteristics and appears to be a relatively large amount of white space spectrum. In a related work [5], I have analysed the capacity of TVWS access technology. The conducted analysis qualitatively describes the main features, strengths, and open challenges of TV white space access

standards and solutions under development. I concluded that TV White Space (TVWS) technology could be involved in designing a general-purpose (in contrast to traffic efficiency applications) wireless communication system for future mobile users. TVWS access strikes a balance between coverage and capacity and can potentially support several thousand mobile users per cell. TV white space technology infrastructure exploitation also represents a viable solution to maximize the use of bandwidth resources and will have a potential impact on the overall need for more spectrum to satisfy the growing demand.

Besides the capacity demands, mobility management in mobile communications has evolved from handling simple and single Radio Access Technology (RAT) handover cases to managing complex, multi-RAT mobility scenarios. To support different levels of Mobile Unit (MU) mobility, future wireless access infrastructure is required to a) Support integration of heterogeneous RATs, the 3GPP communication systems [6] and trusted/untrusted non-3GPP access types, for example, Wireless LAN [7], efficiently combining multiple simultaneous connections of MU via multiple access nodes with unified mobility management b) provide seamless IP mobility for session as well as service continuity as per application needs c) maintaining optimal service level quality for services that have different latency requirements between the MU and the Packet Data Network (PDN), and d) Support optimized mechanisms to control signalling overhead i.e. Minimize the number of messages required to enable the traffic exchange between the MU and the PDN.

Current and upcoming wireless communication systems exploit many techniques to meet these requirements [8]. Wireless Local Area Networks (WLANs) provide high data rates at low cost within a limited area, Cellular systems [9] have achieved wide coverage areas, full mobility and roaming, combining multiple radio resources and deploying a mix of different Radio Access Networks (RANs), frequencies, cell sizes, acquiring new way of deploying, operating and managing multiple RATs that aims to provide high data rate and low latency to mobile users. Furthermore, during recent years, Software Defined Networking (SDN) [10] and Virtualization of network functionalities [11] have gained a lot of attention from the research community and standardization organizations. SDN provides flexible ways to monitor and manage network efficiently with separation of data plane

and control plane. On the other hand, virtualization allows the hardware infrastructure to be provided as a service by abstraction and sharing of physical resources. Both technologies have related mechanisms and therefore could complement one another. Moreover, the appearance of 5G mobile system architecture [12] promises a new connected ecosystem with potential technologies, like Cloud Radio Access Network (C-RAN), Heterogeneous-CRAN (H-CRAN) [13], massive Multiple-Input and Multiple-Output (MIMO) [14], mmWave [15], information (content or data) centric communication, and novel multiplexing. Although the service providers and mobile network operators have started developing industrial solutions for 5G systems, However, there are many dimensions and technologies included in the 5G system that set new requirements for further research. In response to these observations, I set out to take up the mobile network architecture what I refer as S Plus Mobile Network (S+ MN) as a reference model, combined with configurable SDN and NFV technologies to solve many of issues identified in this document. To limit the potentially huge design space and the envisioned research towards automotive access use case, I focus mainly on the vehicular standardization efforts and subsequently analyses the enabling features of current and upcoming mobile networks for vehicular access. It is anticipated that the level of automotive information exchanges enabled by wireless communications will significantly increase in the near future due to a growing number of wireless enabled vehicles [16] and a huge number of applications should be made available to a large number of cars through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications.

Realizing the future automotive environment, I define specific requirements that an optimized access system is expected to fulfil in order to be aligned with vehicular cognitive access described in the present document. I also describe some of the innovative automotive applications of connected vehicles and analyze them to find improvement possibilities and proposed evolved forms for vehicular communications with certain network characteristics (e.g. low latency, high reliability, individual traffic steering, QoS, mobility management, service and/or session continuity, value-added services, etc.). In today's mobile networks multiple services are supported over the same architecture, which would not natively support specified V2X communications. Therefore, specialized network entities and other core network elements should be involved and designed for vehicular messages

broadcasting. A traditional way to achieve a highly customized network is to deploy physical infrastructure for each service (or even one for each business). This approach clearly cannot be applied in an effective way and calls for technical solutions that allow for both efficient resource sharing and multi-tenant infrastructure utilization. In this context, 5G System comes up with several evolutions to provide customized networks and provide optimized wireless communications for specific user class such as network slicing [17], proximity services [18], and mobile edge computing [19] etc. I explored the technical details of these building blocks in the view of assessing for automotive access. There is a general understanding these enhancements will address some of the deficiencies of present vehicular communications standards. Proximity services not only provides a platform for the most desirable safety vehicular communications but also paves the way toward determining the source of autonomous vehicle attacks. Mobile edge computing promises to reduce latencies for some vehicular applications such as the traffic information system that have flexible latency requirements. Network slicing features an efficient and dynamic arrangement of resources to operate as individual networks, thus allowing for massively customizable service [20].

The conducted analysis qualitatively examined the strengths and open challenges of these solutions and established network slicing as the most relevant building block for envisioned customised automotive access system. A network slice can be broadly defined as an end-to-end logically isolated network that includes access as well as core and transport network functions based on designated service requirements. However, network-slicing-based systems can still face challenges caused by the potentially ultra-high density of heterogeneous networks and different points of control [21]. More specifically, in the use case under consideration, the main concern is the high mobility, which includes maintaining QoS, smooth handover, session continuity and sharing across multiple wireless technologies. In cognizance of these challenges, I plan to enhance the S+ MN architecture under network slicing framework proposed for 5G and design a slicing scheme to be generic and adaptable to the specific user class. I also describe the potential functional grouping that must be provided by S+ MN architecture to support specifically configured automotive slice instance.

Furthermore, the proposed automotive slicing architecture will be detailed with exemplary procedures to highlight its key features, such as numbers of signalling messages processed by control entities relative to current mobile network architecture. More specifically, in an intersystem handover scenario, to solve the problem of unnecessary long routes and delay, mechanisms for path optimization need to be developed for network slicing based systems that enable efficient (Re)selection of data plane path during the inter slice handover and the offload mechanism [22]. Traffic has to be distributed with intelligent QoS aware slice management and orchestration mechanisms among different automotive slices, thus ensuring the efficient utilization of network resources and provision of the best service towards the connected vehicles.

### **1.1 Aims and objectives of the research**

The objective of this research is to provide mobile communication network optimisations for automotive user class that demands specific treatment, e.g. in terms of extended coverage, seamless connection mobility, isolation, ultra-low latency and very high reliability. An advanced automotive slicing architecture is proposed for providing tailored support in the slicing-based system and tackling the problem of high mobility while improving the QoS and ensuring efficient utilization of network resources.

The main objectives of the research can be summarized as below:

- Analysis of the current state of heterogeneous mobile networks in the context of vehicular communications and highlighting of the associated design challenges.
- Improvements for SDN and virtualization featured mobile network (S+ MN) architecture to allow the creation of a customized network for providing an optimized solution for a certain user class.
- Collection and management of various types of user-centric, network-centric and context-centric data in unified data server so as to, use the additional information from big data analytics with the system to further optimize QoS and mobility control and management.
- Validating the functional implementation of an S+ MN architecture and to demonstrate how it can coherently executable and implementable.

- Realizing the future automotive environment, identification of specific issues need to be addressed while applying the network slicing scheme for vehicular communications. Define the functional groupings that must be provided by S+ MN architecture to support specifically configured automotive Slice Instance and describing exemplary procedures to further explain the expression of proposed automotive slicing.
- Following the distribution of gateway functions that solve the problem of unnecessary long routes and delay and enable the (Re)selection of efficient data plane paths with implied upper-layer service continuity mechanisms for session continuity.
- Describe an automotive slice instance that provides tailored support for mobility and QoS whilst ensuring the efficient utilization of network resources. The QoS control and management will be flow-based to enforce the required QoS over both wireless access and the forwarding path.
- Develop an experimental design to remove the chains of IP preservation of current mobility management solutions and disburden the process of flow forwarding during IP anchor relocation.

To the best of my knowledge, this is the first comprehensive and profound work on slicing based automotive access. Most of the current research focuses on mobility management in the legacy network of entities using SDN models and have addressed important issues. However, mobility management in a network of functions and the slicing-based network has not yet been implemented.

## 1.2 Thesis structure

Chapter 2 contains definitions of the most important terms used throughout this thesis.

Chapter 3 provides an overview of the state of the art of mobile communication systems and initiatives for architectural evolutions within the 3GPP and IEEE followed by identification of the future research challenges that still need to be addressed.

The common characteristics of the automotive web of services and details the specific requirements for automotive access that an optimized communication system is expected to fulfil are presented in chapter 4. I describe the scenarios addressed by the European ITS communication

architecture from the main fields of applications for the connected vehicle. Then the most related areas of envisioned research in the automotive context are presented. This has the aim of narrowing the potentially huge design space, and to provide the basis towards a feasible solution to fully appreciate the particular challenges carved out in this thesis.

Chapter 5 provides a glimpse of the S+ MN architecture and highlights some of its promising features. With a focus on the customize access solutions that are applicable to vehicular communications, I describe network slicing paradigm from the perspective of mobile network architecture and establish network slicing as the most relevant building block of the 5G Systems. I further elaborate an automotive slicing scheme and related specific issues. Then, I define the potential functional groupings, each encompasses a number of individual functions or applications for individual solutions for certain operational and service requirements. The promising properties of the proposed improvements are illustrated with exemplary procedures of current mobile networks to SDN context. Emphasis is placed on mechanisms for flexible service tailored mobility, QoS and data path optimization.

Chapter 6 evaluates the improvements for mobility control and management, maintaining QoS, smooth handover, session continuity and efficient resource utilization. A functional setup validates the proposed evolution, including its core functionality implemented using the ns-3 network simulator. The benefits achieved in multiple areas are shown with simulation results. Finally, the feasibility of using Multipath TCP for connection mobility in intersystem handover scenario is implemented.

Chapter 7 concludes the dissertation and contains a summary of the achievements as well as the limitations and based on these, directions for further research are proposed.

References and an appendix containing my publications revolving the idea of automotive cognitive access provided to the community are produced at the very end of the thesis.

## 2 Terminology

There are many specific terms defined and used in the context of wireless communication systems standardization that are also used in this thesis. To prevent a misinterpretation of the terms used in this thesis, they are clearly defined in the following.

**Automotive Cognitive Access (ACA):** The ACA introduces a concept of customised mobile network enabling vehicular communication in a heterogeneous radio access environment under higher mobility and QoS constraints. ACA combines the advantages of cognitive computing, ability to obtain knowledge of its current geographical and operational radio environment (available resources, target applications, number of vehicles) and dynamically and autonomously adjust its operational parameters and protocols to operate in an uncertain environment where the access conditions may change from location to location and time to time.

**S plus (S+):** It indicates the evolution of a mobile network system as a whole, as well as individual network entities towards software defined plus virtualization, featured enhancements. In the generic context, S+ also indicates the proposed improvements of mobile system entities and procedures.

**S+ Mobile Network (S+ MN):** It refers to software defined plus virtualization featured mobile network system including S+ Access Network (S+ AN) and S+ Core Network (S+ CN).

**S+ Access Network (AN):** Network elements and functions required to support the radio access operation that includes legacy and novel 3GPP (like E-UTRAN, 5G New Radio System) and also non-3GPP (like WLAN) access technologies. S+ AN is based on the concept of access network as a service. It indicates a set of computing and storage infrastructure resources where part of the AN protocol functionality executed according to the functional split concept investigated in S+ MN.

**S+ AN platform:** Technology baseline implementing one or more specific instances of S+ AN as a service. For example, in case of Infrastructure as a Service (IaaS) based service, the S+ AN platform encompasses the physical infrastructure resources (servers, storage, and their connectivity) and the corresponding management software.

**S+ Access Nodes:** Logical entities introduced in S+ MN system implementing fully or partially the RAN protocol stack. For the



heterogeneous environment, **S+ NB** and **S+ AP** refer to the evolution of the access entities correspond to the implementation of 3GPP and non-3GPP access nodes for operation in the S+ MN system. The terms 5G New Radio (NR) and Next Generation NodeB (gNB) are also used corresponds to NR Base station System according to the specific scenario.

**S+ Access Unit:** Multiple S+ Access nodes can group together to serve aggregated service and are referred to as S+ Access Unit.

**S+ AN service instance:** A specific implementation of an S+ AN platform, serving a set of S+ MN Access node and executing the upper domain of a virtual access node.

**S+ Core Network (S+ CN):** The CN specified in the context of S+ MN system and includes the network functions required to support the backhaul operations. It connects to an S+ AN and allows for a distributed implementation of the forwarding plane.

**S+ Network Controller (S+ NC):** Logically centralized control that allows efficient AN and CN operations such as access nodes coordination, which is particularly useful for addressing inter-cell and intra-cell interference. The terms client context (client specific) and server context (server specific) are used as two types of resource views, the S+ NC works with. A client context represents the necessary and sufficient material in the S+ NC to support a given MU within the same administration that owns the controller. It includes all of the attributes of service as requested by the MU and may contain service-specific information necessary to map service attributes into the realization of the service. A server context is the symmetric counterpart to a client context. It contains everything necessary and sufficient to interact with a group of underlying resources, which could be, for example, a discrete network element or the virtual resources contracted from a partner domain.

**S+ DPE:** Logical entity implementing a subset of the functions of a separate data plane corresponds to 5G UPF. It can be physically located anywhere between S+ NC and S+ AN, or between S+ AN and S+ CN. Each S+ DPE is essentially a transport node operating at a different protocol stack layer depending on the particular functional split, and a set of s+ DPEs is forming a backhaul network whose forwarding plane can be configured by an S+ NC.

**Mobile Unit (MU):** A vehicle connecting/connected to an S+ MN system is referred to as MU. As of this writing, the MU needs to be capable to:

- Indicate support and request for connection mobility for a V2X AS connection as well as IP flow mode indication when a new PDN connection is added.
- Provide handover indication and receive from the network decisions about whether connection mobility applies to a V2X AS connection.
- Exchange routing rules with the network over control plane protocols.
- Route IP flow(s) by using the default access for connection mobility and routing access information which is part of the routing rules.
- Notify the network that access becomes "usable" or "unusable" in static IP anchoring mode.

**Vehicle-to-everything Application Service (V2X AS):** An application function located in the application node providing data service. In other words, V2X AS is a peer node with which a vehicle (MU) is communicating and, on which a corresponding service is running, such as remote diagnostics system, fleet management server, traffic flow optimization and advanced driver assistance applications etc. It can communicate directly with the S+ MN via both control and data plane interfaces.

**Access Link:** Link (DL/UL) connecting the S+ AN and the S+ CN is referred to as access link, e.g. the optical link between S+ AN controller and S+ access node. As a convention in S+ MN, all links within the AN allowing for a distributed implementation of the RF layer of an access node and upper layers will be referred to as access link. The term side link is used for communications between MUs as specified in 5G recent specifications.

**Automotive Slice Instance (ASI):** A specifically configured and instantiated logical network that contains specific access network, core network functions and required resources (e.g. compute, storage and networking resources) to meet certain network characteristics.

**Virtualized Node B (vNodeB):** In 3GPP terminology, a base station is an implementation of a logical radio access network node. This is termed as NodeB in UMTS and eNodeB in EPS (4G system) respectively. Similarly, next generation Node B (gNB) has been named for 5G NR access network node by 3GPP. The S+ AN requires that the functionality of a radio access node can be provided as virtualized AN functions, where each function can be assigned to the slice instance. Using 3GPP terminology this virtualized

implementation of radio access node in S+ MN architecture is named as vNodeB, where prefix v refers to a virtualized instance of a radio access node and NodB represents the 3GPP context. A vNodeB enclosed the set of functions and interfaces correspond to the implementation of a slice instance. S+ vNodeB controller located in the AN service platform is responsible for functional distribution across the vNodeB, consistent execution of the distributed functionalities, management and configuration of the different vNodeB components.

**Handover:** A handover is used to describe when an MU changes its attachment point to a PDN service. The process of performing a handover is referred to as handoff. Handovers can occur between two access nodes that belong to the same technology or different technologies. When the MU handoff between two independent systems e.g., from 3GPP eNodeB to a non-3GPP Wifi AN it is named as inter-system handover, and when the initial and target access technologies are part of the same system it is an intra-system handover.

**Gateway Function:** A gateway function refers to a network node that provides IP assignment (IP anchor) and user plane functionalities.

**Distribution of gateway functions:** The basic concept of distribution of gateway functions is to deploy IP assignment network nodes (IP anchors) closer to a mobile unit. More specifically, as it provides the distribution of IP anchors, a mobile unit no longer uses a single IP address anchored at a central home access network, but it deploys and uses a local IP anchor at visited access network to start new communications. The key challenges of distribution of gateway functions are the selection and re-location of IP gateways while maintaining the reachability of an initial data connection that is still in use by active communications.

**Location Management:** The location management function in the S+ MN Architecture manages the necessary 3GPP mechanisms to track the location of MUs in the S+ AN. The S+MN Architecture introduces new paging and tracking area update procedures aimed at improving the efficiency of standard location management schemes in slicing based networks.

**Traffic steering:** It is the process of distributing V2X service traffic among the available S+ AN and parts of the S+ CN. Through traffic steering, the traffic load can be balanced among access nodes of the same or different

RATs. Traffic steering can be performed statically or dynamically. Policies are generally used for traffic steering, for example, thresholds can be defined which, when a value falls below or exceeds a defined threshold, raises events that result in inappropriate actions. I used traffic steering to minimize the latency, policy control and enforcement in the network by optimizing path lengths and using SDN technology.

**PDN Session:** A PDN session is a concept for an association between the MU and a V2X AS in 3GPP terminology in heterogeneous considered context.

**Session/Service Continuity:** The uninterrupted MU experience of a PDN service, especially the cases where the IP address and/or anchoring point changes. The continuity of a session implies that the IP address is preserved for the lifetime of the PDN session.

**Black and White mobility management:** In today's mobile system (UMTS, 3G and 4G) architecture, the mobility management functions are provided irrespective of the nature of target users. This is referred to as black and white mobility management approach and several issues have been identified by the research community. For example, location update procedures are always conceived even it is not needed in given scenario, leading to un-optimized resources usage and unnecessary delay.

**Seamless/non-seamless Mobility:** The terms seamless and non-seamless mobility apply to the inter-system as well as to the intra-system handover procedures. Seamless mobility is defined as the capability to change the MU's point of attachment to an IP-based network, without losing ongoing connections and without disruptions in the communication. Non-seamless mobility may involve session interruptions during handover. In either case, the establishing/re-establishing a session may also adds overhead. This largely depends on the QoS provisioning strategy. In cases, the traffic is not prone to delay and jitter, non-seamless mobility management is favourable as it can optimise the utilization of access and core network resources. In seamless mobility management, additional network resources are needed as it may include multiple simultaneous connections to maintain the ongoing communication as in the following cases.

**Make-before-break mode:** During a handover with make-before-break mode also referred to as soft handover, the MU can communicate simultaneously with the initial and new access nodes.

**Break-before-make mode:** During a handover, with break-before-make mode, the MU does not communicate simultaneously with the initial and the new access nodes.

**QoS-aware routing:** In mobile communication systems the QoS aware routing refers to the relocation of a PDN-gateway according to application requirement. A PDN-gateway also referred to as an IP anchor is a network node that terminates the user traffic towards an external network and assigns an IP address to the user. During the handover, the relocation of PDN-gateway is often needed as the mobile networks are all IP networks. A handover that is governed by a specific requirement of the MU and V2X application to be fulfilled while handing the connection between IP anchors. In such cases, QoS-aware routing needs to be provided for different IP flow modes.

### 3 Background and State of the Art

This section provides an overview of concepts and initiatives of the current and upcoming architecture for mobile communication systems mainly within the Third Generation Partnership Project (3GPP) [23] and Institute of Electrical and Electronics Engineers (IEEE) [24] that are relevant, extended or applied for S+ MN Architecture.

#### 3.1 Evolved Packet System (EPS)

Evolved Packet System (EPS) represents the Fourth Generation (4G) of mobile networks standard defined by the 3GPP. The overall EPS architecture [25] has two apart components related to technical items being studied by different working committees, known by acronyms: Evolved Universal Terrestrial Radio Access Network (E-UTRAN) intended to the evolution of the radio interface also largely refereed as Long Term Evolution (LTE), and Evolved Packet Core (EPC), which focuses on core network architecture evolution replacing acronym System Architecture Evolution (SAE) [26]. EPS introduced a significant improvement step, characterized by a flat all-IP architecture, an evolved architecture for both the AN and the CN part. Figure 3.1 depicts the overall EPS architecture, including the key network elements and standardized interfaces.

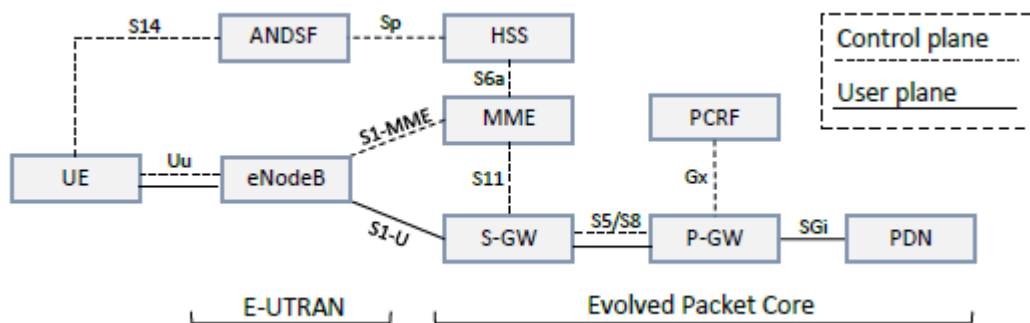


Figure 3.1: EPS architecture overview

*E-UTRAN* is the radio interface of 4G's access network gives the accessibility and access infrastructure to the mobile user in acquiring the EPS network capabilities and services. [25]. E-UTRAN supports up to 100 MHz bandwidth and improvement is achieved by using 8 x 8 MIMO. It is Orthogonal Frequency Division Multiplexing (OFDM) based structure [27]. The E-UTRAN consists of only one network element called eNodeB connected with User

Equipment (UE) and directly connected to CN using the S1 interface. The X2 interface has been defined to inter-connect eNodeBs working in the meshed form. An important function of the X2 interface is to facilitate intra-system handover. There are multiple modes possible by X2-based handovers such as seamless, lossless, selective retransmission, multiple preparations and mobility robustness handling. Depending on the selected mode, the link quality of neighbouring eNodeB is identified that facilitate handover decision. After the handover decision confirmation, the source eNodeB sends a handover request to target eNodeB which include related information. The target eNodeB allocates required radio resources and acknowledge the handover. An X2 GPRS Tunneling Protocol (GTP) connection is established between the source and target eNodeB which carry the user data during the handover. In this way, user downlink data is buffered during handover that reduces packet loss.

The main functionalities of the E-UTRAN can be summarised as follows.

**Radio Resource Management (RRM):** RRM includes functions related to the radio access domain, such as admission control, radio bearer control, link level mobility control, scheduling and dynamic allocation of resources to mobile users in both uplink and downlink.

**Traffic Management:** Together with RRM this helps to ensure scheduled as well as real-time user traffic between the Non-Access Stratum (NAS) [28] of the user side and the infrastructure side. Traffic Management mainly includes the control of different traffic types, activity levels, throughput rates, load balancing. In E-UTRAN traffic management is further enhanced to control the inter cell interferences, transfer delays, bit error rates and effectively maps the traffic attributes used during inter system internetworking to the attributes of the radio access bearer layer of the access stratum.

**Header Compression:** This enforces the compression the IP packet for efficient use of the radio access interface that could otherwise produce a significant overhead.

**Security:** The security related functions make sure that all the data traffic sent over the radio interface is encrypted.

*Enhanced Packet Core (EPC)* is the core network architecture of the EPS. It provides access via different radio access networks and also supports the

mobility between multiple heterogeneous access networks, including 3GPP legacy system like (GERAN or UTRAN) and also non-3GPP systems like Wi-Fi, WiMAX. The main functional nodes inside the EPC are Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (S-GW), Packet Data Networks (PDN) Gateway (P-GW) and the Policy Control and Charging Rules Function (PCRF).

The *MME* is the control node within EPC that provides control signalling between UE and core network and manages authentication, paging, session states and mobility with 3GPP and non-3GPP. MME choose the S-GW and P-GW during attachment procedure and monitors the bearer establishment for transmission of data. It also handled the UE requests for modification and release of bearers. MME also select the targeted MME in handover scenario of mobile from one pool to another pool. The important feature of MME is to provide the data for idle or connected states which later on describe in control and data plane. MME interfaces are; S1-MME with eNodeB, S6a with HSS (for signalling to access mobile data like authentication). S10 interface used for signalling exchanges during UE moves. S11 associated with S-GW interfaces for permitting the establishment of bearer between eNodeB and S-GW.

The *HSS* is a database which contains subscriber and user related information that facilitates in user authentication, access authorization and session setup. It also maintains the information of corresponding PDN service to which the user can connect. This can be in the form of a label according to DNS naming conventions referred to as Access Point Name (APN) of the PDN service or a PDN address indicating subscribed IP address. Furthermore, the HSS holds EPC dynamic information such as the identity of the MME to which the user is currently attached or registered. The HSS may also interact with the Authentication Centre (AUC), which provides the vectors for authentication and security keys. The HSS Packet Data Protocol (PDP) context alteration technique is utilized when the HSS chooses to adjust the subscribed QoS, where ordinarily QoS related parameters are changed.

The *S-GW* is the data plane node connecting the EPC to the E-UTRAN. The S-GW also serves as mobility anchor when UE moves between eNodeBs as well as for other 3GPP technologies. As packet routing node, the S-GW



forwards the data received from P-GW to the eNodeB and vice versa. When eNodeB or P-GW delivered the data to S-GW, it examines the QCI for the purpose of implementation of packet scheduling mechanism. When mobile is in IDLE mode, S-GW initiates the notification to MME for incoming data. S-GW interfaces are S11-with MME, S5 with P-GW (with established bearer between MME and P-GW) and S1-U which is associated with eNodeB.

The *P-GW* being the point of entry and exit of data traffic in EPS provides connectivity to UE through S-GW and terminates the data traffic towards the external PDN service such as the Internet, IP Multimedia Subsystem (IMS), evolved High Rate Packet Data (eHRPD) wireless data networks etc. P-GW is also responsible for allocating IP addresses to connecting UEs, QoS enforcement and flow-based charging according to rules from the PCRF. A UE may have simultaneous connectivity with more than one P-GWs for accessing multiple PDN services. For internetworking with the non-3GPP network it also serves the mobility anchor. P-GW interfaces are s5 with S-GW, Gx with PCRF for implementations of policy rules and SGi with external PDNs that carries the IP packets data.

The Policy and Charging Rules Function (PCRF) is the software node within EPC that enforce QoS policy rules. PCRF takes policy decisions based on the session information received from the corresponding PDN service and user's profile stored in the Subscription Profile Repository (SPR). The PCRF also decides how a service data flow (SDF) shall be treated by the Policy and Charging Enforcement Function (PCEF) and transfer the rules to be implemented to the PCEF [29]. PCRF is also responsible for controlling the flow-based charging functionalities in the PCEF and provides the QoS authorization, QoS class identifier and bit rates that decides how a certain data flow will be treated in the PCEF and ensures that this is in accordance with the user's subscription. The PCRF reference points are Rx with PDN service, Gx with P-GW and Gxc with S-GW.

There are several other network entities extending EPS in different 3GPP releases such as Access Network Discovery and Selection Function (ANDSF) [30], evolved Packet Data Gateway) (ePDG) etc. To limit the potentially huge design space the details of these are not included.

### 3.2 State of the Art

Recently the Fifth Generation (5G) of mobile systems comes up with several evolutions and promises to upgrade the mobile technology to the next level through the realization of large channel bandwidths, spectral efficiency, extended coverage, and full interworking with other wireless communication systems. The evolved 5G system introduced a new connected ecosystem with potential technologies, like C-RAN, Network softwarization, mmWave communications, Mobile Edge Computing (MEC), Information (content or data) centric communication, and novel multiplexing. Furthermore, 5G system enables different user services classified into three main categories, namely, massive Machine Type Communication (mMTC), enhanced mobile broadband (eMBB) and Ultra Reliable Low Latency Communication (URLLC) [31]. In the following, I first introduce the 5G reference model with an overview of the defined Network Functions (NFs) and service discovery/selection principles in the 5G system. Next section describes the end-to-end protocol layering of 5G system. Finally, I discuss the 5G enhancements for optimal resource usage, mobility management and QoS Provisioning.

### 3.3 5G System Architecture

Figure 3.2 shows the 5G System reference model, which is a generalised design of the functionalities of the network architecture [12]. The architectural elements are defined as network functions that offer their services via two distinct interfaces representation between these network functions: The service-based interfaces and reference point interfaces. The service-based interfaces represent the interaction between network functions with the 5G Control Plane (CP) and connecting the control plane from Radio Access Network R(AN). The reference points will continue to exist within the R(AN) and User Plane (UP) functions and are numbered from N1 to N10 and so on. The various architecture figures can be found in [12]. Figure 3.2 shows one of these, which is for a non-roaming scenario, including the key network functions and the standardized interfaces.

*5G-Access Network (AN)* is the air interface of 3GPP's upgrade path for mobile access network that implements the infrastructure and accessibility to the user in acquiring the 5G network capabilities and services. 5G New Radio (NR) [32] deployments are planned predominantly into two

Frequency Ranges, FR1 for frequency bands between 410 MHz – 7125 MHz, notably it was below 6 GHz in the release 15, and the FR1 range is being extended to 7.125 GHz to allow NR to be used for unlicensed bands in the 6 GHz region in release 16. The FR2 range defines for mmWave bands, in the 24 GHz, 28 GHz, and 38 GHz ranges. Also, there have been 3GPP study Items on extending FR2 to include higher frequencies, including 60 GHz. The 5G NR base station is also referred to as next generation NodeB (gNB) in some documents [33]. A number of multi-carrier (MC) and single-carrier (SC) waveforms have been proposed for the 5G air-interface [34]. 5G-AN with the common characteristics that it connects to 5GC supports one or more of the following possibilities: 1) Standalone NR, the gNB connects to the 5GC. 2) Non-Standalone NR with E-UTRA extensions. 3) Non-Standalone NR in EPS, NR using the existing LTE radio 4) Standalone E-UTRA access systems. The given 5G-AN options are identified to serve different 5G deployment scenarios from eMBB to uRLLC and to mMTC. Key technological enhancements of 5G-AN include ultra-lean transmission, low latency communications, advanced antenna techniques, spectral efficiency with operation in upper frequency bands, inter-cell radio resource control, Radio Bearer (RB) management, radio admission control, dynamic resource allocation and inter-working between upper and lower frequency bands. The uplink carrier will avail Cyclic-Prefix Orthogonal Frequency Division Multiplexing (CP-OFDM), as well as Discrete Fourier Transform (DFT), spread OFDM waveforms. Unlike E-UTRAN there will be a Peak to Average Power Ratio (PAPR) difference on the uplink signal [35]. The standalone 5G NR has an increased channel bandwidth up to 100 MHz and wide spectrum allocated in new bands such as the n77 NR band has a 900 MHz wide frequency range from 3.3 GHz to 4.2 GHz. Furthermore, three-dimensional beamforming will be used to achieve higher network capacity and higher data throughputs in new frequency bands. Using these technologies, however, changes the radio access from cell coverage to beam coverage, representing a significant change from 4G Radio Access Networks (RANs).

It provides access via different radio access networks and also supports the mobility between multiple heterogeneous access networks, including 3GPP legacy RAN like (E-UTRAN) and also non 3GPP systems like WLAN access. The main functional elements within 5GC are Access and Mobility

Management Function (AMF), Session Management Function (SMF), Policy Control Function (PCF) and Unified Data Management (UDM).

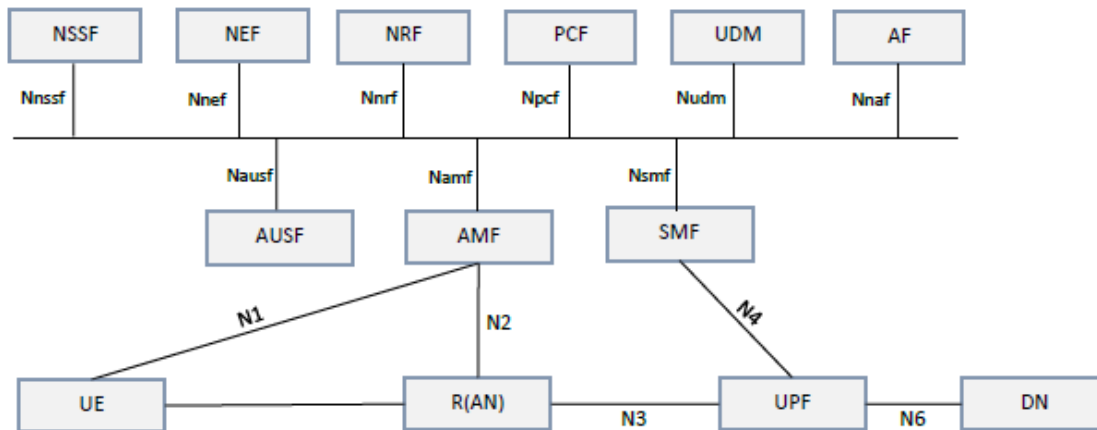


Figure 3.2: The generalised design of the 5G System Architecture [12]

The AMF includes the functionalities of RAN CP interface termination, access authorization and authentication, NAS signalling termination, NAS ciphering for integrity protection, registration and connection management [36]. The most important function of AMF is to provide the mobility control and management that includes location services management, transport of location services messages between UE, R(AN) and Location Management Function (LMF), transport of session messages between UE, SMF and Short Message Service Function (SMSF). The AMF also allocate EPS Bearer ID for interworking with EPS and select the target AMF in handover scenario of mobile from one access pool to another. AMF interfaces are; N1 with UE, N2 with R(AN) for signalling to access mobile data like authentication. Namf service-based interface exhibited by AMF allows an NF consumer to communicate with the UE and/or the R(AN) to provide related services such as subscribe or get notified for mobility related incidents and statistics.

The SMF mainly provides the configuration of traffic steering for UPF for routing of traffic which includes IP address allocation and management, selection of UP function and session establishment. In addition, in certain cases, the SMF is responsible for maintaining tunnel between UPF and R(AN) node, session modification and release. The SMF provides termination of interfaces towards policy and charging rule function, handle local enforcement to apply QoS, policy and charging data collection and support of charging interfaces, control and coordination of policy data collection at UPF. Furthermore, SMF can also provide support for interaction with

external data network service for the transport of related signalling in cases such as data session authorization or authentication by external data service.

*PCF* is responsible for policy rules provision to control plane functions. It provides a unified policy framework support to regulate network behaviour. The *PCF* may request and provide policy information to the subscription data server and implements an anchor to access subscription information needed for policy decisions in the User Data Repository (*UDR*).

*UDM* includes two parts *UDM* Front End (*FE*) and *UDR*. *UDM-FE* supports processing services such as subscription management, access authorization, handling of user identification and location management. The *UDM-FE* enforces the service logic and does not require an internal user data storage. The *UDR* provides storage functionalities for data required during *UDM-FE* processing, plus policy profiles required by *PCF*. The data stored in the *UDR* includes user subscription data, including subscription identifiers, security credentials, access and mobility related subscription data, session related subscription data, policy data and so on. The *UDM-FE* accesses subscription information stored in a *UDR* to supports its functionalities.

In addition to these main network functions, there are some other specified and optional functions introduced in 5G Systems, such as Authentication Server Function (*AUSF*), Network Slice Selection Function (*NSSF*), Network Exposure Function (*NEF*), Network Repository Function (*NRF*) and Application Function (*AF*). *AUSF* that enables unified authentication between the UE and 5G CN in the initial attachment procedure. The Network Exposure Function (*NEF*) translate information from external service to internal network and events and expose internal capabilities to the external application. In order to support the application influence on traffic routing, accessing *NEF* and interacting with the policy framework for policy control, the Application Function (*AF*) is introduced to interacts with the core network. The Structured Data Storage Function (*SDSF*) and Unstructured Data Storage Function (*UDSF*) are optional functions that support the saving and retrieval of information by the *NEF* as structured data and as unstructured data respectively.

The user plane of 5G systems is provided via *User Plane Functions (UPFs)*, that support various data path operations and services such as packet routing and forwarding operations, service detection operations, packet inspection, QoS handling for data plane, policy rule enforcement, bitrate enforcement operations, the anchor point for inter or intra RAT mobility (where needed), traffic accounting and reporting etc. The user plane QoS in 5G Systems is flow-based, the UPF inserts QoS marking (a scalar value) in the packets and packet flows of the same QoS marking belong to the same QoS. In addition to QoS handling, the packet inspection and packet forwarding and routing are supported. The UPF also acts as PDN session point of interconnect anchor point to external data service. The number of UPF deployed in the user plane path is not constrained in 3GPP specifications and also the separation of control and data planes provides further deployment flexibility. Based on UPF deployment scheme centralized or distributed, the selection and re-selection of an appropriate UPF and supporting a subset of optional functionalities is managed by the SMF according to the required UE functionalities and capabilities. UPF interfaces are; N3 with R(AN), N4 with SMF, N6 with a Data Network and N9 between different UPFs. More details of respective roles of each of these functions are described in the next sections.

### **3.4 Protocol Stack in 5G**

The 5G protocol stack is almost like EPS with some additions/changes according to related functional enhancements. The protocols between the UE and the 5G-AN/5GC elements, between the 5G-AN and the 5G core network functions, or within the 5GC network functions are categorized into user plane protocols and control plane protocols. Control plane protocols control the connection setup and management from various aspects which includes service request, different transmission resources control, handover signalling etc. This also includes the mechanism for transparent transfer of NAS messages [28]. User plane protocols enforce the active data session services which carry user data through the access stratum [37].

#### **3.4.1 Control Plane Protocol Stack**

5G Control plane protocol stack comprises of NG Application Protocol (NG-AP) [38], Stream Control Transmission Protocol (SCTP) [39], NAS protocol for MM functionality (NAS-MM), NAS protocol for SM functionality (NAS-SM) and set of 5G-AN layer protocols depending on the 5G-AN implementation

options. The service-based interfaces use Hypertext Transfer network Protocol HTTP/2 with JavaScript Object Notation (JSON) as the application layer serialization protocol.

### 3.4.1.1 Control Plane Protocol Stacks between the 5G-AN and the 5GC

**5G-AN – AMF:** The Control Plane Protocol Stacks between the 5G-AN and the AMF over the reference point N2 is depicted in Figure 3.3. For every function in AN, the exact mapping is done in AMF through N2. NG-AP protocol provides the signalling service between the access node (5G NR, gNB) and the AMF, a single NG-AP protocol is used for both the 3GPP access and non-3GPP access [40]. NG-AP consists of Elementary Procedures (Eps) of AMF Configuration Update, RAN Configuration update, initial context setup, handover resource allocation, cancellation and preparation and so on. Like TCP, the SCTP protocol defined in [39] ensures reliable in-sequence transport of messages with congestion control. SCTP is also message-oriented like UDP that provides redundant paths between a 5G-AN node and AMF.

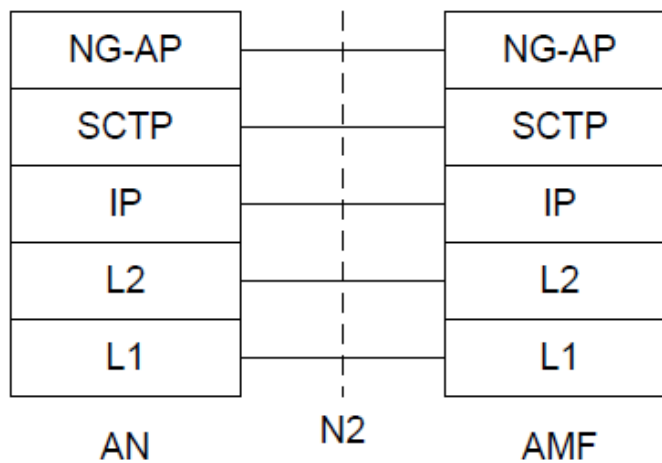


Figure 3.3: Control Plane protocols between a 5G-AN and AMF over reference point N2

**5G-AN - SMF:** The control plane between AN and SMF as shown in Figure 3.4, exist in the presence of subset which eases the functionality giving a transparent relay to send and receive messages. This subset acts as a relay reference point between AN and SMF. For the purpose of decoupling between AMF and SMF and in case, the SMF needs to manage the services provided by 5G-AN, the N11 related messages such as Mobility Restriction, provisioning of QoS requirements that are distinctly forwarded by AMF between the 5G-AN and the SMF. Procedures related to PDU sessions and handover management are also encountered in UE context management.

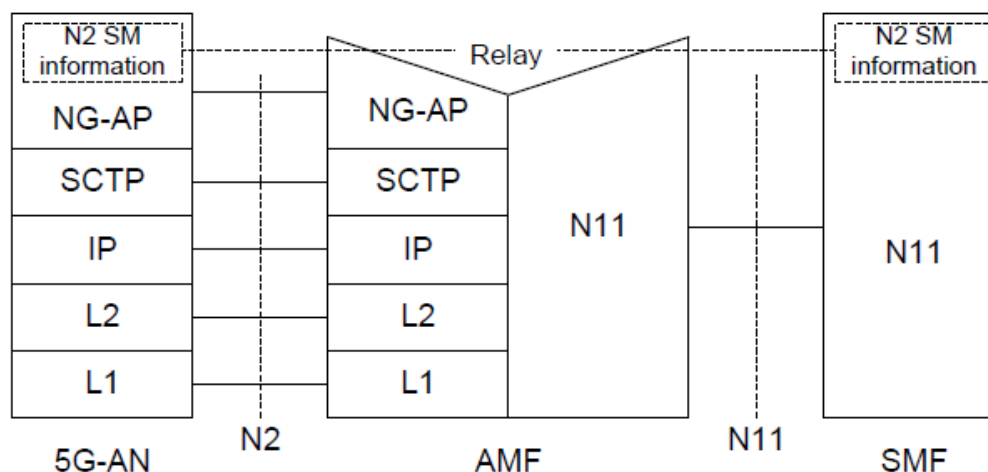


Figure 3.4: Control Plane protocol stack between a 5G-AN and a SMF [12]

### 3.4.1.2 Control Plane Protocol Stacks between the UE and the 5GC

To access any connection inside UE, a single N1 NAS signal is used. This signal is transmitted for both Registration Management (RM) and Connection Management (CM). There are two basic components of the NAS protocol namely, NAS-MM and NAS-SM [41]. Some of the other protocol between UE and a 5GC transported over N1 via NAS-MM are SMS, UE policy, Location Services (LCS) and Session Management Signaling. It is also possible to signal another type of NAS message together with an RM or CM NAS message. Protocol security of the NAS message is provided based on the context link between UE and the 5GC.

**UE – AMF:** Control Plane between the UE and the AMF is responsible for connection and mobility management control signalling. The NAS protocol is found on the N1 reference point between the UE and AMF. This signalling includes operations such as registration, paging, security mode control, authentication and configuration updates (NAS-MM). In addition, NAS signalling between the UE and the AMF is transported over the N2 control plane interface. As shown in Figure 3.5, the 5G-AN serve as a relay protocol layer between UE and AMF that transfer UE context, resource and mobility management support messages. Following the NAS message, the AMF sends authentication and key agreement request. This signalling includes information regarding access control and session management procedures.

**UE – SMF:** Control Plane between the UE and the SMF (Figure 3.5) forms a complex set of handling that exist in the presence of NAS-SM. NAS-SM



handles signalling of SM in the relay layer between the UE and the SMF. On the other hand, NAS-MM layer in this control plane handles the transmission and reception of SM. Transmission of SM eases in security header and reception of SM performs the integrity check to derive the control plane between the UE and the SMF.

NAS-SM: The NAS protocol for SM functionality supports user plane PDU Session Establishment, modification and release. It is transferred via the AMF, and transparent to the AMF. 5G NAS protocol is defined in the specification for access to the 5G System via non-3GPP access networks [42].

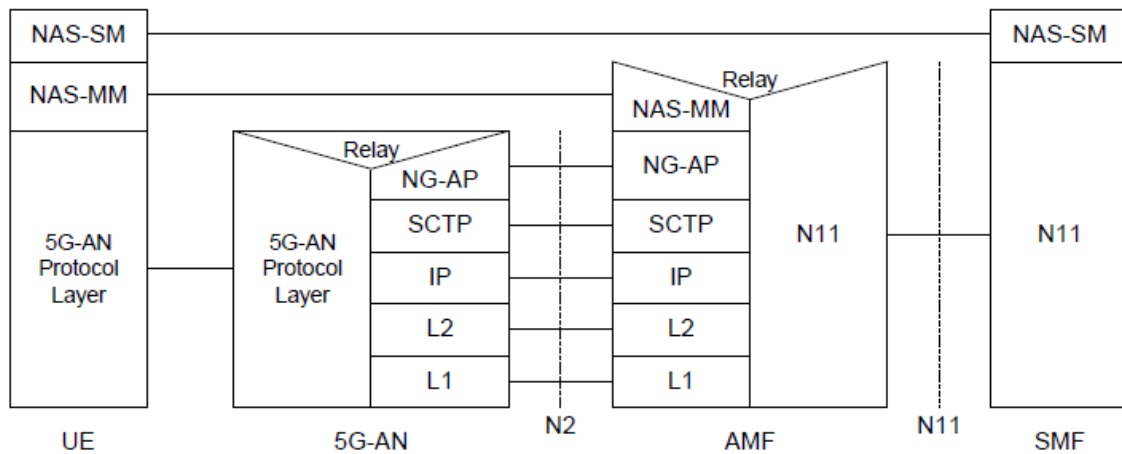


Figure 3.5: Control Plane between a UE and a SMF

### 3.4.2 User Plane Protocol Stack

Figure 3.6 illustrates the protocol stack for the User plane, implementing the actual PDU Session service. *5G-AN layers protocols* depend on the AN (3GPP NG-RAN or Trusted/Untrusted non-3GPP access) and are defined in [42] and [43] respectively. The radio protocol between the UE and the 5G-AN node (5G NR base station/gNodeB) are Radio Resource Control (RRC), Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), Medium Access Control (MAC), Physical (PHY) and Non-Access Stratum (NAS) sub-layers [44]. The 5G-AN use the same protocols, except for the Service Data Adaptation Protocol (SDAP) [45]. The SDAP has been introduced in 5G for flow based QoS, as described in the following sections. It provides the mapping between QoS flows and data radio bearers and marking QoS flow ID (QFI) in both DL and UL packets [32]. UDP/IP are the backbone network protocols.

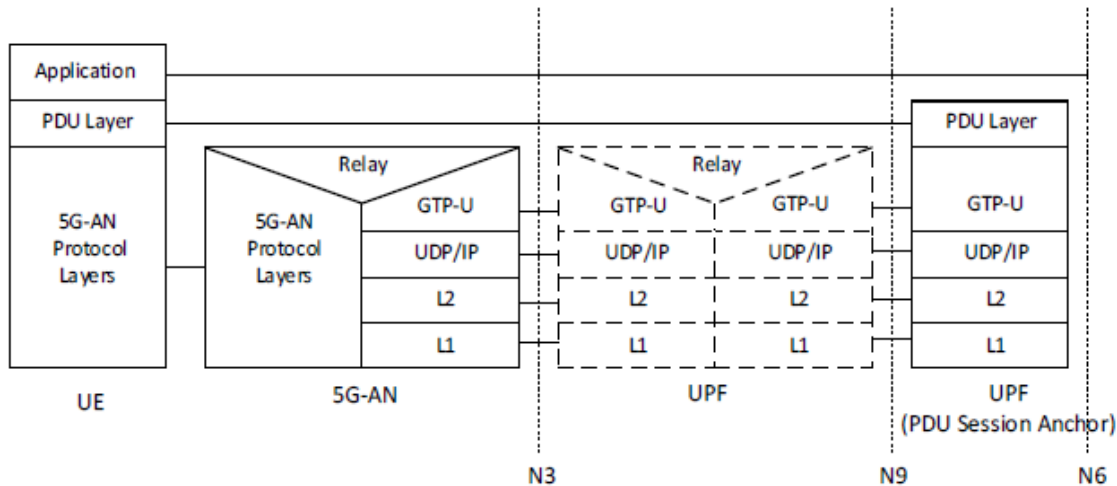


Figure 3.6: User plane end-to-end protocol stack [42].

The number of UPFs in the data path may be zero, one or multiple in the data path of a PDU Session. The *PDU layer* corresponds to the PDU carried between the UE and the PDN network over the PDU session. The IPv4 or IPv6 or IPv4 PDU Session type corresponds to IPv4 packets or IPv6 packets or both of them. When the PDU session type is Ethernet, it corresponds to Ethernet frames etc. GPRS Tunnelling Protocol for the user plane (GTP-U) supports multiplexing traffic of different PDU Sessions (possibly corresponding to different PDU Session Types) by tunnelling user data over N3 (i.e. between the 5G-AN node and the UPF) and N9 (i.e. between different UPFs of the 5GC) in the backbone network. GTP provides encapsulation on a per PDU Session level of all end user PDUs. This layer carries also the marking associated with a QoS Flow.

### 3.5 Mobility Enhancements in 5G System

Mobility control and management is an imperative component for any new wireless standard, particularly when the connected users move to a new access point that belongs to different wireless technology. The fourth generation of mobile networks standard and prior wireless communication systems make use of vendor specific hardware and software to implement mobility related functionalities. This has been identified as an inefficient approach by research community during recent years. In response to these observations, the 5G system architecture specification [12] indicates a complete departure from non-flexible ways to manage the mobile network

with separation of data plane and control plane. As shown in figure 3.7, the mobility management functions in 5G Systems are mainly performed through the AMF block, Multiple 5G New Radio (NR) base stations form cell-based as well as beam-based coverage depending upon 5G-AN deployment option as described in section 3.3. Multiple NR base stations can be connected to single or multiple AMF and form the Tracking Areas (TAs). Multiple TAs are further grouped into TA Lists (TALs). During attach procedure the AMF identifies the mobile user current location in the TAL through paging procedure. When a UE moves out of the current TAL, the

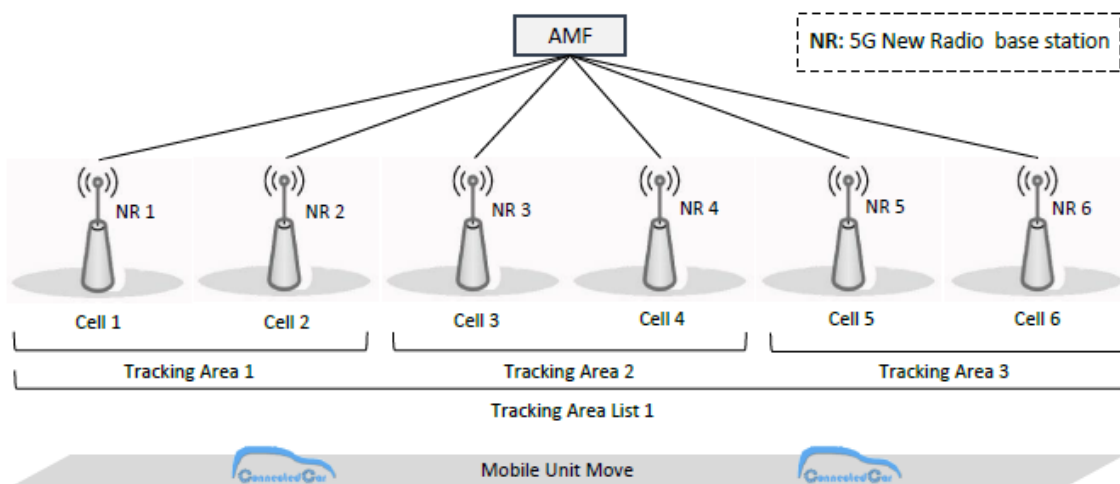


Figure 3.7: 5G access and mobility framework

UE reports its new location to the network through the location update procedure. The AMF may sequentially page a cell, the TA of the cell, and/or TAL of the cell.

To appreciate the storms of black and white mobility management approach in current mobile networks (i.e, mobility management procedures are always provided even not needed in some cases), the 5G system defines three different modes for mobility management for UEs referred to as Session and Service Continuity (SSC) mode 1, 2 and 3 [12]. The SSC mode 1 is similar to classical IP mobility management, for the session continuity the IP address is preserved. The SSC mode 2 provides no session continuity support, IP address is also released when the UE changes the point of attachment. The SSC mode 3 support service continuity when session continuity is not provided i.e., the application level mobility management. The IP address is not preserved in this mode as the IP anchor is changes. The network ensures that the UE does not loss connectivity with new

connection establishment before the release of the initial connection. For location tracking based solution, the 5G system enables the network to provide adaptive management of TA list assignment and a mobile UE can connect to multiple RATs, while having application data flows with different QoS criteria. Instead of the TA list, a cell list is maintained in the CN and transmitted to the UE via NAS signalling. This allows to assign individual cell lists to each UE, whereas in the TA based solution, all UEs in the same TA use the same cell list. Thereby each UE can use its specific area with a finer granularity than TAs for paging and location updating. This also implies that the area boundary is different for each UE. Therefore, one particular enhancement is that signalling overrun of many UEs changing TA at the same time can be avoided. With the cell list solution, it is not necessary to manage Tracking Area Identity (TAI) on the RAN level, i.e. no TAI is broadcasted by the RAN nodes.

In addition to radio connection establishment/maintenance between UE and RAN, the 3GPP enhanced the RRC/RRM to includes the mobility and QoS management functions. Furthermore, it also allows the implementation of such RRC and RRM in the CN. Consequently, the AMF block can control the RRC/RRM functions. These enhancements provide additional benefits, For example, this enables the NR base station to be a purely data plane entity and any interaction between the RRC/RRM and AMF blocks can be executed at the same physical location, and hence reduce the control plane signalling between the access node and the core network.

To enable the 5G system to reconstruct the state when the UE is connected and sends small data without having to continuously maintain state in the network between instances of connectivity, the concept of a Context Cookie is introduced in the 3GPP architectural study for next generation of the mobile system [8]. The Context Cookie is assigned by the 5G system to the connecting UE and once the UE has been successfully authenticated and connected to the network this context information is re-used by the 5G system in other related procedures. Based on network policies the context cookie is applicable depending on the device type and/or service type the UE is connecting to. With this solution, the UE can perform small data transfers without requiring the UE to transition to the connected mode and without any mobility management signalling. The UE can also use the Context Cookie to transition from idle to connected without requiring the

network to maintain a full context for the UE. After a period of inactivity during which the network can release the UE context.

To appreciate the highly heterogeneous and ultra-dense expected nature of 5G mobile systems, the delivered 3GPP stage 2 level release 16 specification [46] has described the handover signalling not only for the 5G system itself, but also handover signalling mechanisms for legacy networks (intrasystem) such as 4G LTE, 3G, 2G and non-3GPP networks (intersystem) like WLAN access. To implement an intersystem handover, which are more challenging due to the interaction of two fundamentally different access systems, 3GPP has specified an interworking architecture [12] as shown in Figure 3.8. In the illustrated interworking scenario between the 5G network and EPS, only the interfaces that hold relevance to the description in this thesis are mentioned. The EPC and the 5GC are connected to a set of interworking elements via the E-UTRAN/MME for the EPC and NG-RAN/AMF for the 5GC respectively. Further, the EPC through its existing interfaces also allows user mobility from 5GC to 2G and 3G networks [12]. The illustrated interworking scenario uses an N26 interface between the MME and the AMF. This interface although essential for interworking between the legacy and 5G networks, the support of N26 interface in the initial deployment phases is optional until a full-scale rollout of 5G system due to multiple reasons such as longer adoption times, capital expenditures etc. For internetworking between 5G System CN and EPC, the PCF + PCRF, SMF + PDN GW Control functionality (PGW-C) and UPF + PGW user plane functionality (PGW -U) are defined. These dedicated functional groupings are optional and are based on UE subscription and mobility management capability. Where 5G system and EPC internetworking does not need the UEs may be served by own entities of the respective network, i.e. either by P-GW/PCRF or SMF/UPF/PCF. If needed as for distributed mobility management, there can be additional UPF between the gNB and the UPF + PGW-U. Alongside new handover signalling methodologies the 5G System enhanced user's mobility event tracking and reporting functionality. The AMF enables the user mobility related event reporting to SMF, PCF or NEF that has been authorized to subscribe this service. In service-based communication procedures, if service consumer network function subscribes to the UE mobility event notification service for reporting of UE presence in a given area, the AMF tracks user's location related to its state and determine the presence of UE in the given area using AN procedure as described in [46].

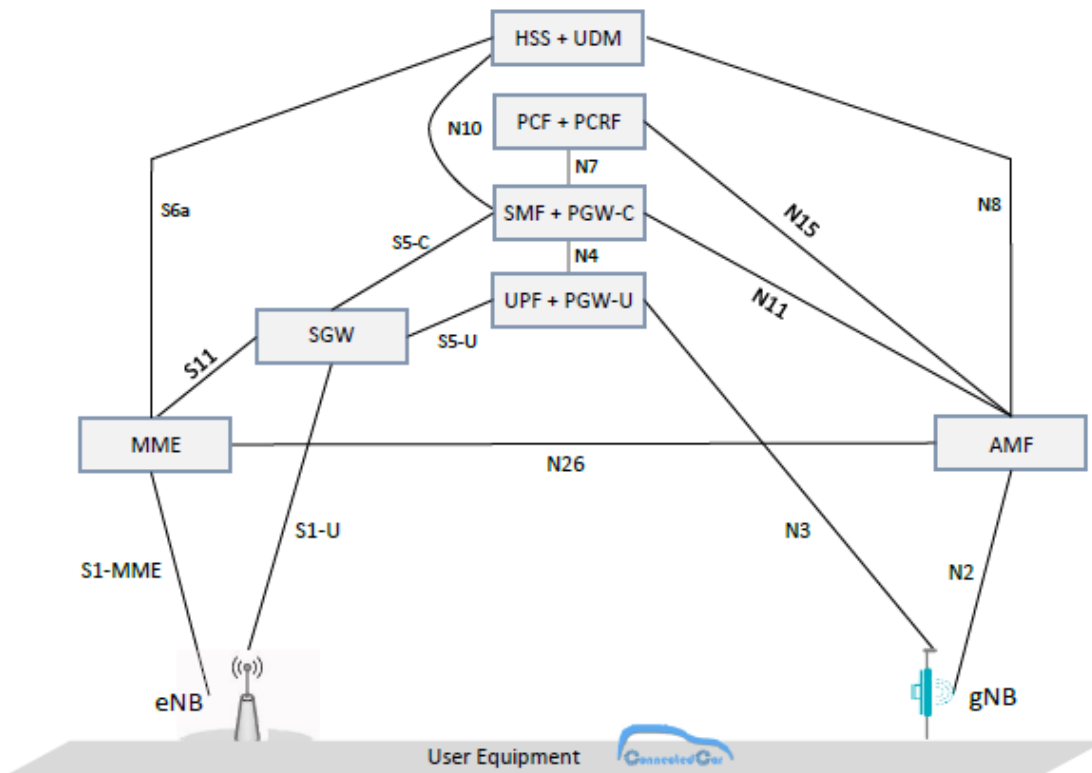


Figure 3.8: Non-roaming architecture for internetworking between 5GS and EPS

As the change of the UE's presence in the given area is detected, the AMF notifies the UE presence and new location to the subscribed service consumer network function. In case the AMF is changed, the subscription of mobility event reporting is transferred from the old AMF based on mobility context of the UE. The new AMF may decide about the continuity of current status notification related to subscription of mobility event to corresponding network function. In addition, it is introduced that the AMF to use the user mobility patterns based on subscription to further optimise and personalise the UE mobility [47]. This enables to define and update the UE mobility statistics, network local policy. The statistics of the mobility of UE also includes area update timers per user and historical or expected UE moving trajectories.

The proposed enhancements address many of the mobility related problems, however, there are many emerging protocols and technologies included in the 5G system that raised new research questions. The future containing the challenges and problems of mobility management which is targeting the 5G of mobile networks are discussed in a later section 3.7 to round off the research.

### 3.6 Quality of Service (QoS) Provisioning

QoS in 5G Systems is provisioned through “QoS Flows” that replaces the concept of “bearer” and a QoS flow corresponds to the user plane traffic within a PDN session. One or more service data flows (SDFs) of similar QoS characteristics are grouped together in a QoS flow that receives the same QoS treatment. Each QoS flow is characterised by a QoS profile which includes 5G QoS Identifier (5QI) and Allocation and Retention Priority (ARP) parameters. Any QoS profile also has one related QoS Flow Identifier (QFI) that is not included in the QoS profile itself. The identified QFI provides information to network elements to which a flow to map the service data flow. 5G QoS characteristics (Resource Type, Priority profile, Packet Delay Budget (PDB), Bit Error Rate (BER), Maximum Data Burst Volume (MDBV), Averaging window) associated with 5QI provide the packet forwarding treatment that a QoS Flow holds between the UE and the UPF [12]. Table 1 summarizes the standardized 5QI to QoS characteristics mapping.

*Table 1: standardized 5QI to QoS characteristics mapping (adopted from [12])*

<b>Example Services</b>	<b>5QI Value</b>	<b>Default Priority Level</b>	<b>Packet Delay Budget</b>	<b>Packet Error Rate</b>
Conversational voice	1	20	100 ms	$10^{-2}$
Conversational (live) video streaming	2	40	150 ms	$10^{-3}$
Online gaming, V2X communication messages, Medium voltage distribution network, Automated process monitoring	3	30	50 ms	$10^{-3}$
Non-Conversational (buffered) video streaming	4	50	300 ms	$10^{-6}$
Mission Critical user plane push to talk voice (MCPTT)	65	7	75 ms	$10^{-2}$
Non-Mission-Critical user plane push to talk voice	66	20	100 ms	$10^{-3}$
Mission critical video user plane	67	15	100 ms	$10^{-3}$
IMS signalling	5	10	100 ms	$10^{-6}$
Video TCP-based (Buffered Streaming) (e.g., World wide web, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)	6	60	300 ms	$10^{-6}$

Voice, Video (Live Streaming), Interactive gaming	7	70	100 ms	$10^{-3}$
Video TCP-based (Buffered Streaming)	8	80	300 ms	$10^{-6}$
World wide web, e-mail, chat, ftp, p2p file sharing, progressive video	9	90	300 ms	$10^{-6}$
Mission critical delay sensitive signalling (e.g., MCPTT signalling)	69	5	60 ms	$10^{-6}$
Mission critical data (example services are the same as TCP-based video (live/buffered) streaming.	70	55	200 ms	$10^{-6}$
V2X messages	79	65	50 ms	$10^{-2}$
Low Latency eMBB applications, Augmented reality	80	68	10 ms	$10^{-6}$
Discrete Automation (Delay Critical GBR)	82	19	10 ms	$10^{-4}$
Discrete Automation	83	22	10 ms	$10^{-4}$
Intelligent transport systems	84	24	30 ms	$10^{-5}$
Electricity distribution high voltage	85	21	5 ms	$10^{-5}$

The QoS profile identifies the 5G QoS characteristics with given QFI and ARP parameters that determine a Priority Level (PL). The ARP values determine which flow takes precedence and can also be used for flows admission control. Like EPS, 5G QoS supports both Non-Guaranteed Bit Rate (Non-GBR) and Guaranteed Bit Rate (GBR) classification. In addition, 5G introduce delay critical GBR classification for ultra-reliable low latency communication services. These classifications are part of the QoS profile and are defined at the QoS flow level. Certain QoS rules are associated with each QoS flow which can be a combination of packet filters for detecting traffic and corresponding QoS profile. For example, to limit the overall usage across all SDFs within a QoS flow, an Aggregate Maximum Bit Rate (AMBR) is provided for Non-GBR flows. Similarly, a Guaranteed Flow Bit Rate (GFBR), a Maximum Flow Bit Rate (MFBR) and a maximum packet loss rate values are provided for GBR flows. Furthermore, a flow descriptor is introduced to describe the packet treated. It is composed of a Flow Priority Indicator (FPI) and Reflective QoS Indicator (RQI). FPI refers to parameters which are preconfigured at AN node for packet treatment and RQI indicates



whether UE shall apply reflective QoS for corresponding flows. FPI and RQI are used by UPF for transport marking over NG3.

5G enhances the QoS control within the network and specified the distribution the QoS functions of the current 3GPP architecture between the UE, AN and CN [8]. Figure 3.9 illustrates a high-level view of such functional split.

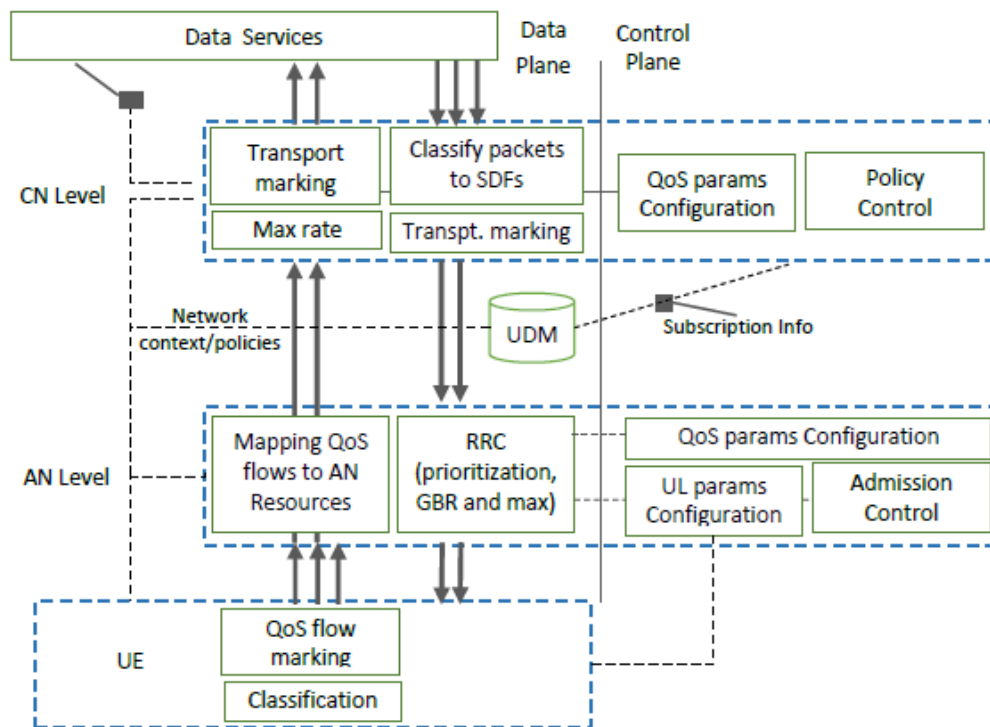


Figure 3.9: 5G QoS functional split, classification, marking and mapping to AN resource

At the UE level, the classification and marking of user plane traffic is performed by UE based on QoS rules. There are three scenarios of assigning QoS rules to QoS flows:

- Pre-authorized QoS rules: UE receives QoS rules for certain QoS flows during PDN session establishment. The targeted QoS flows are identified by the packet filters and UE will insert the QoS marking (a scalar value) in the UL packets and the RAN also receives the corresponding QoS marking over N2 interface and treat those packets accordingly.
- Reflective QoS: The 5G QoS model supports Reflective QoS in case there is no explicit signalling for configuring QoS rules. As reflective QoS can only be used for non-GBR flows, the QoS to be applied on the DL traffic is decided by network and insert QoS markings in the

DL traffic. When the UE receives the DL traffic it derives a new QoS rule from reversed DL IP headers and QoS marking is the same as that in DL packets. However, the mapping between QoS marking and QoS metrics is still not standardized, a UE's PDN session may consist of multiple QoS flows and it allows that some flows are reflective and others are not. Which QoS flows should be reflective can be signalled by the network during PDN session establishment. It is also possible that for the same flow a pre-authorized rule is provided, and it is also chosen to be reflective, in this case, which rule will have higher priority remains an open issue.

- Default QoS rule: if a QoS flow is neither chosen to be reflective nor provided the pre-authorized rule, it may use a default QoS rule which may be provided by the network during PDN session establishment.

At the AN level based on the operative QoS parameters for the session and flows, the access and admission control function determine the settings on which the UE establish a request for a connection in the random-access channel. The admission control also regulates the flows admission in the access network in case of resource scarcity. This also includes to sacrifice already admitted flows to allow more prioritized flows. Furthermore, the resource control function at the AN level distributes the access resources based on the authorized QoS parameters from the QoS Operator control function. This also monitors the fulfilment of the QoS targets. The resource management functions are different in 3GPP and non-3GPP ANs with regards to available options to control resource utilization and availability. The Figure 3.9 also illustrates the 5G QoS flows user plane traffic marking and mapping to AN resources. There are overall two steps of mapping. First step is the NAS level mapping between a flow to a proper QoS marking and the second step is the mapping of the QoS flow (flow with a QoS marking) to the Data Radio Bearers (DRBs). The AN can also configure via RRC the UL mapping between QoS flow and DRB. The first step can be performed either through default QoS rule, pre-authorized or reflective QoS rule. For flows assigned to be reflective, the UL QoS flow are mapped to the same DRB that carries the DL flow. For flows that are not reflective and no RRC configured mapping is available, the flows will be carried over default DRB. The traffic shaping and policy enforcement in the AN is also bound by Intent level QoS rules received from the CN control functions that may refer to this marking.

At the CN level, the SMF controls the PDN session and associated QoS flows. Based on information provided by the PCF, QoS rules are determined and QFI and QoS profile is assigned to the flow. Then the SMF provides the UPF with the Packet Detection Rules (PDRs) for SDFs mapping to the QoS flows. The SMF also sends the following information to the UPF to enable QoS aware flow classification, bandwidth enforcement and marking of data plane traffic [12].

- QoS related information such as MBR for an SDF, GBR and MBR for a GBR QoS Flow.
- The related packet marking information (e.g. the QFI, the transport level packet marking value such as the DSCP value of the outer IP header.
- A DL Packet Detection Rule (PDR) enclosing the set of SDF template for DL packet filter and PDR for UL enclosing the set of SDF template for UL packet filter.
- In case of Reflective QoS, optionally an indication of reflective QoS for the DL PDR.

Additionally, to ensure the coherency between UPF and (R)AN, the SMF also transfer the QoS rules to (R)AN. This ensures that the UPF and (R)AN are working in a coordinated manner. The R(AN) is bound by the charging and detection performed in the core and is responsible for providing an appropriate QoS to the flows marked by the core network. The SMF may also set up an explicitly signalled QoS rule and provides it to the UE together with an add operation [12]. To summarize, the 5G network control functions (PCF, SMF and AMF), the access node, the UPF, and the UE all work together to enhance QoS provision and how the QoS is characterized and enforced. Following enhancements are recognised compared to the EPS QoS framework:

- 5G enhanced the QoS control within the network with QoS flow which is the finest granularity of QoS handling.
- The synergy between the network and service layer is increased that enabled the service layer to demand certain admission, retention and notification behaviour from the network. This approach provides an improvement in the network response as compared to currently available admission control distinction between GBR and non-GBR

PDU flows. Moreover, with this approach, the network monitors the QoS realization and notify the service if the targeted QoS cannot be achieved.

- QoS parameters are enhanced as compared to EPS QoS model. For example, the Required Bit Rate parameter is introduced for any kind PDU flow irrespective of "GBR" or "non-GBR" type. Furthermore, the authorized QoS parameters per PDU flow are aligned with application requirements parameters per SDF.
- 5G enhances the QoS options, such as the Option for explicit QoS targets such as delay, bitrate, etc. can be specified and enforced. Similarly, to achieve targeted explicit QoS the delivery characteristics per flow are decoupled from PDU flow priority.

### **3.7 Evaluation and Open challenges for 5G Network**

This chapter provides the related work which presents the background and state-of-the-art in the research area of mobile communication systems. A brief introduction to 3GPP's 5G system architecture is presented, highlighting some of its main characteristics. By 2022, global mobile devices will grow from 8.6 billion in 2017 to 12.3 billion and over 422 million of those will be 5G capable [1]. Research and development work on 5G system is ongoing and 5G services are envisioned to provide connectivity for mMTC, eMBB and uRLLC. The Key Performance Indicators (KPIs) for these different categories are defined by ITU [48]. As described above, different new technologies are being applied in the 5G system to address challenges in the field where research is placed, namely capacity enhancement, resources management, mobility control and management, the policy and charging control capabilities, traffic steering and application oriented QoS provisioning. Millimeter wave frequencies are included to use in hotspots and indoor areas to offer high spectrum efficiency and throughput [49]. To alleviate backhaul and core network and to allow executing delay-sensitive and context-aware applications in the proximity of users, mobile edge computing is used by moving computing, storage, and networking resources to the edge of AN [50]. These enhancements promise to i) increase the spectral efficiency by a factors of five to fifteen compared to current networks; ii) extend the access density ten times higher, i.e., at least 106/km<sup>2</sup> to satisfy the demands of massive connectivity; iii) satisfy the requirements of a low latency (radio latency 1 ms); iv) support diverse compelling services and diverse variety of usage scenarios or use cases

[51]. In line with these directions, although the service providers and mobile network operators have started developing industrial solutions for 5G systems, However, there are many dimensions and technologies included in 5G system that set new requirements for further research. In the following, I describe the identified open challenges, which needs to be addressed while implementing 5G system architecture.

**5G radio access:** One of the key challenges of 5G radio access that needs further research efforts is the implementation of the abundance of new operating bands that has been specified by 3GPP. This implies that 5G radio access systems have to deal with more fragmented spectrum, more complex design with new ways of deploying and operating the radio access techniques. Another major challenge is the level of interaction required between 5G NR and legacy E-UTRAN until a full-scale rollout of NR stand-alone deployments [52]. In particular maintaining uplink with different networks in non-standalone situations as reported during discussions at IEEE 5G World Forum 2018 [53]. Furthermore, with the cloudification of 5G radio access, many base stations bundled together in the cloud. Despite its great advantages, such cloud based access network architecture imposes challenges on the network scalability and also has a huge risk of a single point of failure. New access schemes will be needed to handle a massive number of non-orthogonal users (more than 10 devices/m<sup>2</sup>) in an efficient and scalable way [54].

**Mobility management and Multipath communications:** The introduced 5G mobility framework promise to provide improved mobility management with multipath support. As described earlier, to support different levels of Mobile Unit (MU) mobility, 5G system access infrastructure is enhanced to a) support mobility management signalling for user registration to the network, reachability to enable mobile terminated communication, detection of MUs no longer reachable, assignment of control plane and user plane network functions, the definition of mobility states and how to transition between the states, support of geographical location services and mobility restrictions, e.g. forbidding mobility at certain locations; b) support mobility on demand for different mobility levels over a given area within a single RAT or combination of RATs, in the service area of a control plane or user plane entity; c) mobility support in interworking and network migration scenarios etc [36]. However, with the standardization and development of

5G, the access systems move from cell coverage to beam coverage [32], that implied new ways of maintaining connection mobility. For example, for certain handover cases, mobility aware early resource reservation mechanism is proposed to minimize the end-to-end latency. However, when a lot of MUs are handed off between adjacent access unit of heterogeneous nature and the large volume of location related signalling will be simultaneously generated for and cell reselections. In this case, the complexity of resource reservation is obviously increased such as for vehicular networks. Moreover, the propagation delay of the warning messages needs to be further minimized for use cases that required frequent handovers [55].

Another consideration is the multipath communication support in order to address the specific mobility needs of different applications and services. To appreciate the related challenges 5G system introduced mechanisms to obtain the information (e.g. application's needs, device capabilities, used services) for certain session requirements, prediction aware forwarding path establishment and support different levels of data session continuity based on this information. However, in order to allow a rich routing environment that is adversely affected by the location of distributed gateway functions is still a key challenge for 5G networks and needs further optimization algorithms to allow full use of available paths in the network. Furthermore, these enhancements have not yet been implemented in the network of functions and slicing-based systems. Within a slicing based network, the capability of flexibly supporting mobility of users and their connected MUs, as well as sessions and flows and even (virtual and physical) network entities is predominantly seen as a challenge in terms of changing performance. More specifically, in vehicular networks, how to realize mobility related event reporting, the path management with seamless connection mobility, minimize the signalling, define the Tracking Area List (TAL) and allocate them to connecting vehicles etc. while sharing across multiple wireless technologies.

**Cognitive resource management:** New methods of dynamic resource distribution techniques are being introduced within 5G instead of using static resource allocation. However, due to limited network resources, increasingly diversified network services and the continues boom of applications and services, it is challenging to efficiently provision network

resources to network elements. Especially with the advent of composite radio environment of 5G networks, efficient and flexible resource allocation schemes with interference awareness are needed [56]. Moreover, the envisioned virtualized cloud of access and core network elements will have the advantages of physical resource pooling but, new schemes will be needed to handle the complexity and difficulty of virtualised resource management among different network slices. Another consideration is the capability to aggregate resources from multiple technologies and frequency bands. For example, to overcome bandwidth scarcity issue FR1 frequency band range is being extended to 7.125 GHz to allow NR to be used for unlicensed bands in the 6 GHz region in release 16. Even though the use of vacant frequencies at a given time in a given geographical area not being used by licensed services (termed as white spaces) has been identified as an important spectrum resource, however, the provision of unlicensed bands also imposes extra limitations that reveals a distinct lack of synergy between introduced enhancements and practices. The introduction of TV white space would require some changes in the specification of given use cases, some amendments are necessary to the current standard documents and architectures. For example, TV white space access technology role in ITS reference architecture [5].

**Emerging applications challenges:** 5G intends to deal with large categories of use cases and fully satisfy their applications requirements. However, the unprecedented expectations about the levels of QoS the 5G system will be able to provide appear very challenging for current 5G technological development. Moreover, some emerging applications such as autonomous driving, the tactile internet, industrial automation and safety critical communications will push the required latency, outage probability and network intelligence to deliver predictive actuation and senses to extreme levels [57]. For example, future vehicles are not only transport tools but also information and entertainment centres for passengers. Many different services need to be provided by 5G vehicular networks. Hence, the massive wireless traffic is expected to increase for 5G vehicular networks. It is a great challenge to improve the service efficiency for 5G vehicular networks.

**QoS provisioning challenges:** The 5G QoS enhancements promise to ensure that all mobile internet users receive sufficient resources and

subsequently receive an acceptable QoS. However, for certain user groups like vehicular communications that do not use a single service at a given point in time, and they are instead operated in an ever-changing environment comprising of many different concurrent services, there are aspects of using static policies and providing priority to specific data flows over coexisting services that have not yet been addressed in a totally satisfactory manner. Given the different set of QoS characteristics by the diversity of the V2X applications and traffic, in conjunction with their increasing expectations, the task of QoS provisioning needs to be approached from the perspective of real-time information about the capabilities, service types, states, location etc. More specifically, in cases such as in an inter system handover where simultaneous access to a single or multiple V2X application services via heterogeneous access networks to be provided, the data flows undergo multi-RAT handovers and each flow's handover require careful RAT and access node selection, IP packet forwarding, and/or route optimization methods. Henceforth, it is clear that the QoS aware routing will be critical during inter-system handover permitting seamless mobility to the connected vehicles. Furthermore, the enhancements of QoS differentiation in the PDU Session frequently consider the question of QoS Flows finest granularity of traffic forwarding treatment (e.g. scheduling, admission threshold), but rarely consider provisioning resources from the perspective of application-layer QoS. In addition, improving the marking algorithms of traffic forwarding treatment is predominantly seen as a challenge and the network should apply different QoS settings for different applications. Finally, as briefly mentioned above, it is foreseen that network slicing will be a key ingredient of 5G system that replaces the QoS profiles used in LTE and UMTS, Therefore, these enhancements need to be realized to support the implementation of a customised QoS in slicing-based mobile networks.

**Testing for 5G:** To apply various advanced technologies and to evaluate the new 5G applications, effective testing methods are needed in a real, rapid and flexible manner. However, the testing methods are faced with new challenges along with the continuous development of the new 5G technologies [51]. Such as the testing of antenna systems, the impacts of all the channel characteristics need further study through channel measurement and channel modelling, in particular for massive MIMO and mmWave channels [52]. Similarly, the field test needs to evolve to validate



the networking performance of many different types of expected services with the evolution of 5G radio access from cell coverage to beam coverage. The test solutions should further evolve to have the ability of testing contents of different new technologies, making measurements Over The Air (OTA) and monitor the live data traffic in emerging coverage scenarios.

To summarize, currently 5G solutions are in the key technologies verification stage and there are many open challenges, which needs to anchor while deploying the 5G network. Some possible fact which will arise during full-scale 5G rollout and can impact the and development of 5G are as follows.

**System Intelligence:** The volume of data exchanged by mobile users will continue to increase and exponential growth in the number of devices connected is expected. To cope with increased data traffic, the future mobile communication system will become further intelligent, with cognitive learning mechanisms to autonomously adjust its operational parameters and protocols based on the user's experience. Mobile system's intelligence will be used to allow fast and situation-aware networking.

**Increasing coexistence:** RATs continue to evolve to include more frequency ranges, wider channel bandwidths, and fast spectrum reallocation with consequent large bitrates available to the users. Infrastructure densification with the usage of low, medium, and high dense cells to increase capacity and enable better exploitation of the spectrum. More specifically, the availability of higher large bands beyond 100 GHz (D-band, 110 - 170 GHz and onwards) can be allocated for high data rate transmission systems over short distances. The lower band radio spectrum with higher propagation characteristics can particularly be specified to extend connectivity in those scenarios where networks suffer from limited radio range and pervasive roadside communication infrastructure.

**Wireless as a service:** With distributed wireless enabled systems everywhere, users will not necessarily need to bring a smartphone but will benefit from wireless devices as a service. All information being in the cloud, users will just need to be authenticated and then access the network by using any available device.

**Vertical service scenarios:** Many vertical industries will move onto the 5G network, to support various service scenarios from these verticals that have very different network capability and performance demands. Every

single scenario requires a different access service and poses requirements that are different. These vertical service scenarios indicate that future networks need to be more flexible and scalable to support customised connections of diverse performance requirements into a single network infrastructure. Moreover, multi-tenancy management will be required to provide a high degree of customization to support each target vertical service needs.

**Energy management:** The energy consumption for the network elements as well as end-user devices has to be reduced. Energy usage will considerably be optimised in different ways such as the need to put devices on recharge will be reduced and that the battery life will be further extended specifically for massive machine type communication. In some scenarios, wireless energy transfer may possibly be a viable way of energy management.

**Quantum Networking:** We are currently witnessing efforts towards quantum networks, with the appearance of quantum computers, capable to work out the challenges that are not solvable with classical computers. For example, the evolution of IoT and the Industry 4.0 will bring the network very close to the real infrastructures and a security breach on the networking side may quickly become a more critical challenge in the real life. The Quantum Key Distribution (QKD) enables secure cryptographic key exchange communication whose security relies on quantum mechanics. Consequently, quantum networks will be available for communications and will force to re-think the cryptography and security mechanisms.

## 4 Design Selection

In this section, the specific areas of envisioned research within mobile network architecture are presented. This has the aim of narrowing the potentially huge design space, and to provide the basis towards a feasible solution to fully appreciate the particular challenges carved out in this thesis.

### 4.1 Automotive Access

Automotive Web of Services or Connected Cars describes an important group of mobile users, where a huge number of emerging applications should be made available to a huge number of vehicles. Connectivity and appropriate infrastructure to interoperate automotive software platforms are the foundation for future vehicles to offer enhanced user experience and evolve the activities associated with driving towards an Intelligent Transportation System (ITS) [58]. Figure 4.1 illustrates a conceptual view of the future automotive environment where various automotive applications need to communicate with each other by wireless connections.

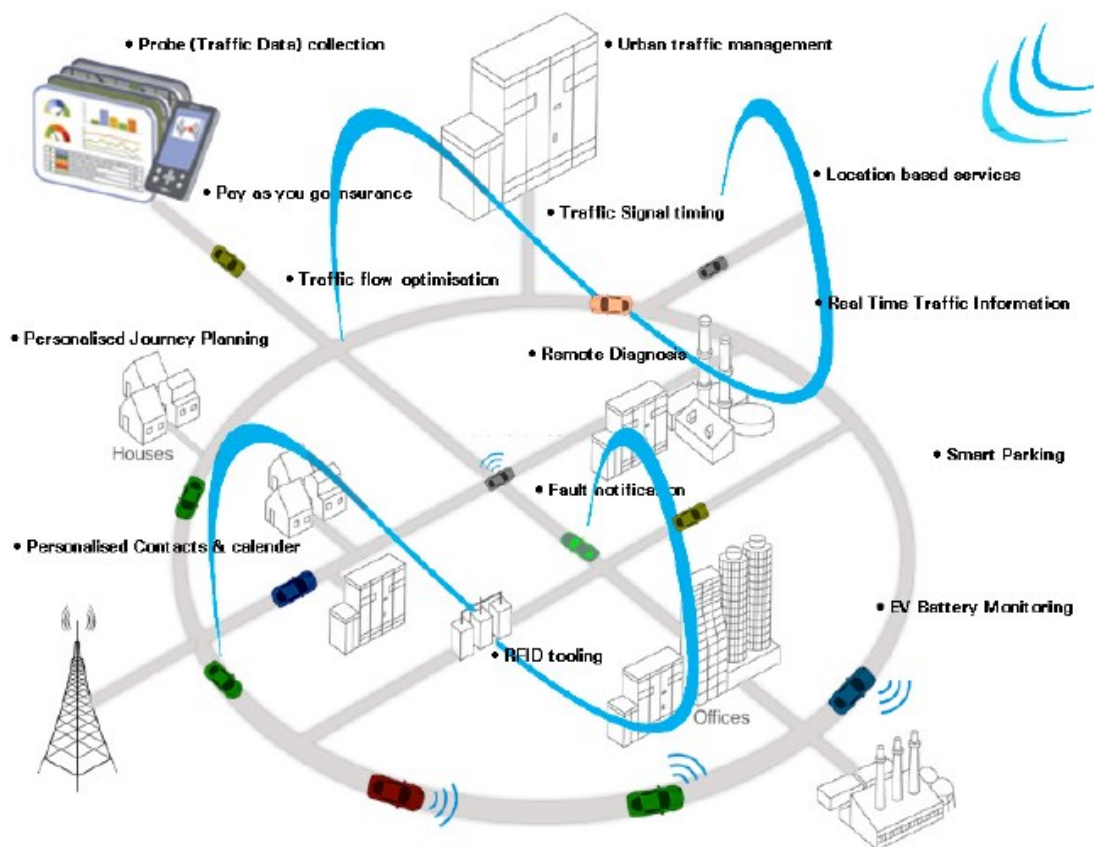


Figure 4.1: Conceptual view of future automotive environment

Realizing the future automotive environment ITS organizations identifies four types of vehicular communication modes: Vehicle-to-Network (V2N) Vehicle-to-Infrastructure (V2I), Vehicle-to-Vehicle (V2V) and Vehicle-to-Pedestrian (V2P) [59] and are collectively called as Vehicle-to-everything (V2X) ecosystem in [59] and its enhancements (eV2X) are introduced in [60]. V2V mode includes vehicular MUs direct wireless communication with other vehicles and V2P mode covers vehicular MUs wireless access to Vulnerable Road Users (VRUs), such as pedestrians, bikers, motorcyclists, and wheelchair users. V2I puts vehicular MUs in communication with roadside infrastructure, for example, Road Side Units (RSUs).

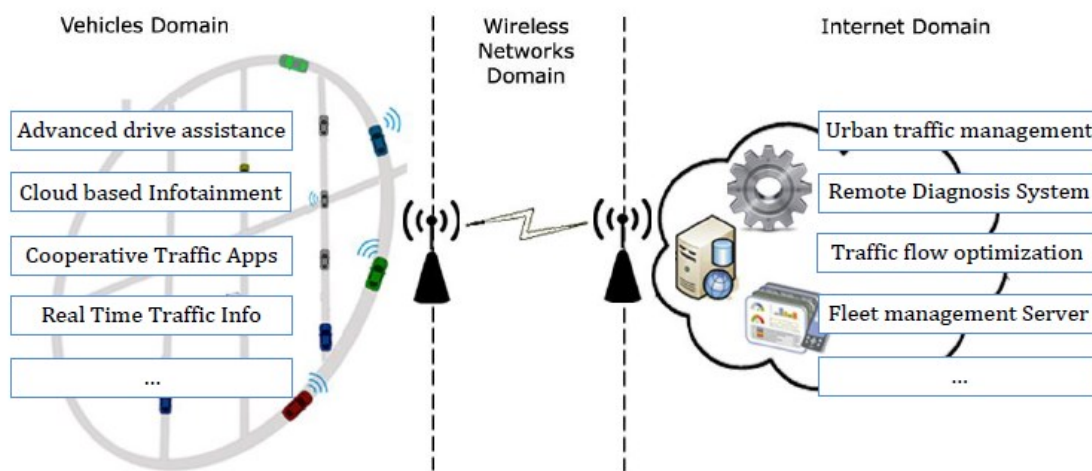


Figure 4.2: V2N context applications

V2N puts vehicular MUs in communication with a centralized backend information system on the network side application server supporting these applications. The V2N includes periodic collection of information by vehicles, from internal and external sensors (e.g. CAN bus, in-vehicle camera, environmental monitoring sensors) and transmission of this information to V2X AS on the network side. The backend system collects data from vehicles for further processing and analysis. This real-time information is combined with other existing traffic data such as traffic management centres, automated vehicle location systems, mobile devices, and connected vehicle equipments, resulting in the up to date, complete and reliable traffic information and provided back to vehicles. Based on the underlying application this system could be further connected with service delivery platforms, enterprise applications or service providers. Figure 4.2 shows a conceptual view of the end to end wireless communications and innovative ITS applications of V2N class considered in this contribution.

## 4.2 Vehicular Communication use cases

The scenarios addressed by the European ITS communication architecture [61] is from the following main fields of applications for connected vehicles.

**Traffic efficiency and safety:** V2V and V2P periodic or event driven notification messages represented as Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs) in ETSI documents [61], [62] support the applications such as forward collision warning, cooperative adaptive cruise control and Vulnerable Road Users (VRUs). These Basic Safety Messages (BSMs) carry the position and range of movement parameters of the connected vehicle by sensing the surrounding environment and notifying the driver about potentially hazardous encounters with other vehicles or VRUs. Example scenarios where these safety messages are generated include ambiguous visibility in a queue or under foggy conditions, pedestrians crossing the road, vehicles approaching a crossing in a built-up area and a curve in a road etc. Sensor data from cameras or radar can be used to detect possible collision. Furthermore, providing drivers individual instantaneous advice about approaching traffic light intervals, general traffic events and congestion alert to enable traffic efficiency as well as fuel economy gain. These application needs continuous and real time exchange and processing of vehicles position, velocity and status data and can extend to the domain of other useful applications.

**Tele-operated driving:** In tele-operated driving, the vehicles driving tasks are taken over by an operator physically remain outside the vehicles. By using live camera view, status, and sensor data, a remote driver controls given vehicle in areas that are either dangerous or uncomfortable to human beings, such as the locations of nuclear plants, an earthquake and snow-ploughing. Connection to the vehicle needs to be obtained via extreme real-time communication (tactile internet) to ensure fast and fully reliable vehicle control. The onboard virtual driver system acts on the inputting commands from a remote operator and interacts with the vehicle collecting sensor data. The onboard platform with read and write access to the Controller Area Network (CAN-BUS) also enable the local processing of sensor data to reconstruct the model of the surrounding area and transfer the only information needed to the teleoperation control system, thus avoiding communication overload [63].

**Autonomous driving:** Also, referred to as self-driving that enables a vehicle to watch/sense its surrounding environment and moving with little or no human input. Requirements in self-driving are stricter than those in tele-operated driving and V2X safety applications. In some scenarios, for example, an autonomous driving vehicle may drive very close to other vehicles at higher speeds (up to 200 km/h) and the data exchange over V2N links may further challenging for autonomous driving efficiency and safety. A self-driving vehicle requires full road network coverage to be driverless in all geographies, with the network supporting very secure and reliable V2N communication.

**Vehicular Internet and infotainment:** This use case provides the connected vehicle with internet access for web browsing, social media access, files/apps download, and HD video streaming etc. The vehicular internet and infotainment for passengers have already established as a must-have feature for new vehicles and would become even more relevant with envisioned penetration of autonomous driving vehicles, in which the driver may also be involved in internet consumption.

**Remote diagnostics and management:** A V2X service provided by a car manufacturer or a vehicle diagnostic centre that enable remote interaction with a vehicular onboard system. This service retrieves information periodically from the vehicles in V2N mode to track their status for remote diagnostic activity [59]. Similarly, fleet management applications may track the vehicle status and position for forensic diagnostic purpose.

#### 4.3 Vehicular Communications Requirements

Vehicular communications have certain unique features, in terms of access primitives, traffic patterns, delivery requirements, and spatial scope. I defined the following common requirements that an optimized communication system is expected to fulfil in order to be aligned with the future automotive environment described in the present document.

**Support of large number of vehicles:** With growing numbers of connected vehicles it is anticipated that a single access unit needs to serve a very large number of connected vehicles [64]. Sufficient area capacity and appropriate communication network should provide a mechanism to

reduce peaks in signalling and the data traffic resulting from large numbers of vehicles, almost simultaneously attempting data and/or signalling interactions. When the network is in overload, it should be capable to provide a mechanism to restrict downlink data and signalling as well as access towards a specific access node. There must be an optimized resource management procedure.

**Extended coverage:** Since vehicles have to travel across different regions and given V2N applications need to communicate with corresponding servers wherever they are thus requiring widespread network coverage. In addition, due to the always changing location of vehicles and the corresponding variance in the access channels, it is needed that the wireless communications network capable of adapting the signal strength to the actual location.

**Mobility Management:** Mobility management is the most challenging requirement for vehicular communications, as vehicles are moving objects with high relative speeds up to 200 km/h and above. It is very likely that vehicles move away from the coverage area of the connected access network and need to change their points of attachment. To support different levels of vehicular mobility, future wireless access network is required to a) support efficient, customised paging and tracking area update procedures. b) handover management among heterogeneous access networks, this requires mechanisms to connect with multiple different access networks and efficient handoff schemes to provide flawless mobility. c) provide connection (IP) mobility for the session as well as service continuity supporting a variety of ITS applications at different changing locations.

**QoS Provisioning:** Each of the automotive use cases described earlier has its own set of services and related wireless communications requirements on the access system. Table 2 summarize the classification of ITS applications and their preferred communication specifications. In traffic efficiency applications, the decentralized Floating Car Data (FCD) services require periodic transmissions of collected sensor data by the onboard systems (e.g., from CAN bus, internal and external cameras, environmental monitoring sensors etc.) to the remote servers [65]. The remote management servers evaluate collected data, process and send up-to-date traffic information back to the vehicle's onboard systems such as predict

traffic congestion and suggests an alternative route. These applications do not have a very wider range of QoS requirements, but their service gracefully degrades with increases in packet loss and delay. Whereas automotive safety applications require very strict reliability and minimum QoS parameters. Tele-operated driving use case requires extreme real-time communication to ensure fast vehicle observation and control [66]. Requirements in autonomous driving are stricter over a wider range of parameters than those in V2V safety applications and tele-operated driving. For many new automotive applications, the vehicles also need to engage in media consumption or in situation the transmission of the real-time video data from the front vehicle to rear vehicle and data exchange over V2N links further enhance the autonomous driving efficiency and safety. Due to the fact that the transmitted video is expected to be used in real times, image-processing and feature extraction in an uncompressed format to avoid encoder delays with aforementioned reliability is a very challenging task [66]. Given the diversity of the V2X applications and traffic, in conjunction with their increasing expectations, the task of QoS provisioning needs to be approached from the perspective of real-time information about the capabilities, service types, states, location etc. More specifically, in cases where simultaneous access to a single or multiple V2X application services via heterogeneous access networks to be provided, the data connection undergoes multi-RAT handovers and require careful RAT and access node selection, IP packet forwarding, and/or route optimization.

**Ultra-low latency and very high reliability:** The International Telecommunication Union (ITU) has categorized major V2V services, such as autonomous driving, traffic safety and tele-operated driving as URLLC services [48]. Inter vehicle cooperative awareness message must have a maximum tolerable end-to-end latency of approximately 3.3 ms [48]. Similarly, very high reliability comes from the field of certain traffic scenarios such as in cooperative adaptive cruise control (also known as platooning) that enables a group of vehicles to travel on the same path, reliability is a key requirement. The probability of an accident will rise if retransmissions are required in a braking needed situation. All trucks must be informed about the forehead condition and brakes must be applied at the same time and with the same negative acceleration.



Table 2: ITS applications and their preferred communication specifications

ITS Applications	Communication type	Optimized communication specifications
<b>Traffic Efficiency</b>		
<ul style="list-style-type: none"> <li>• Urban traffic management</li> <li>• Traffic flow optimisation</li> <li>• Priority for selected vehicle</li> <li>• Smart Parking</li> <li>• RFID tooling</li> <li>• Queue warning</li> <li>• Traffic Signal timing and optimal speed advisory</li> </ul>	vehicle-to-roadside unit, vehicle to sensors (V2I) and V2N: High penetration rate. Vehicle average speed 20, 40, 60, 80, 100 km/h	Latency: Low (100ms) Data rate: Not a concern Reliability: Medium QoS support: Yes
<b>Autonomous driving and Traffic Safety application [67]</b>		
<ul style="list-style-type: none"> <li>• Real-time Traffic Information</li> <li>• Do Not Pass Warning</li> <li>• Forward collision warning</li> <li>• Blind intersection</li> </ul>	V2V and V2I: High vehicle speed Minimum radio. Interference Safer: Towards zero road accidents	Latency: ultra-low latency (1ms) Data rate: high data rate (10 Mb/s downlink) Reliability: High reliability (nearly 100%) QoS support: Yes Explicit
<b>Tele-operated driving [68]</b>		
<ul style="list-style-type: none"> <li>• Real-time navigation</li> <li>• Cooperative adaptive cruise control &amp; platooning</li> </ul>	V2V and V2I: Highly secure Guaranteed transmissions of high-priority Messages Minimum radio Interference	Reliability: High reliability (up-time) Latency: ultra-low latency (1ms) Data rate: high data rate (10 Mb/s downlink) QoS support: High
<b>Cloud based infotainment and internet [69]</b>		
<ul style="list-style-type: none"> <li>• Personalised Journey Planning</li> <li>• Online office</li> <li>• Media playlist</li> <li>• Contacts &amp; calendar</li> <li>• Email &amp; news</li> </ul>	V2N: High vehicle speed	Latency: Medium latency Data rate: High data rate QoS support: Yes
<b>Remote Diagnosis and management</b>		
<ul style="list-style-type: none"> <li>• EV Battery Monitoring</li> <li>• Fault notification</li> </ul>	V2N: Long communication range Small data bursts Low error rate	Latency: Medium latency Data rate: Not a concern Reliability: Not a concern QoS support: Yes
<b>Value-added Services</b>		
<ul style="list-style-type: none"> <li>• Location based services</li> <li>• Pay as you go insurance</li> <li>• Freight and Fleet Applications</li> <li>• Probe (Traffic Data) collection</li> </ul>	V2I and V2N: High vehicle speed Large Communication range Broadcast message capability	Latency: Low (100ms) Data rate: Not a concern Reliability: Medium

**Small data bursts:** As vehicles also need data transmission with wireless sensor networks and many vehicles would send extremely small amounts of data at a time. For instance, in [70] the packet size of around 50 bytes is endorsed for the optimal average delay in mobility scenarios. Increasing the packet size in the vehicular communications, the average information delay to reach the dissemination area increases. Therefore, the vehicular communication system must be able to efficiently handle the transmission of such small data bursts in machine type communications.

**Reliable:** One essential requirement is that the message or data delivery must be reliable.

**Secure:** With the growing threat of hacking and unauthorized compromising of systems, security is an issue high on the agenda of many users.

**Broadcast message capability:** There could be instances where broadcast messages may be needed. The ITS system must be able to accommodate this type of message

**Service Continuity:** Last but not least an important requirement for many of the aforementioned ITS applications is to provide seamless IP mobility for the session as well as service continuity as per application needs.

#### 4.4 Heterogeneous environment

The current trends for radio communications systems indicate a composite radio environment (Figure 4.3), where multiple different RATs links may be available at the same time from legacy 3GPP technologies such as GSM, GPRS, LTE to recent technologies such as LTE-Advanced, 5G AN as well as wireless local area networks may be available at the same time. Different types of radio access nodes stimulate a mixture of femto, pico, micro and

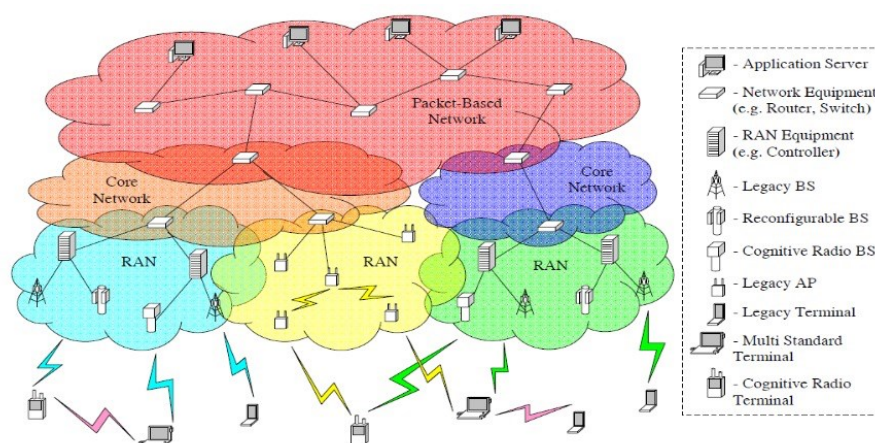


Figure 4.3: Heterogeneous environment

macro cells to provide wireless access to mobile users in this environment. Examples are legacy Base Stations (BTs), WLAN Access Points (WAPs), multi-standard evolved access nodes and 5G New Radio with E-UTRA extensions etc. Moreover, different types of wireless enabled devices operate in this environment such as legacy terminals, smartphones, connected vehicles, smart meters, internet gaming consoles, surveillance cameras, smart TVs, healthcare monitoring, transportation, and package or asset tracking devices.

This hybrid radio environment and predicted a huge increase of traffic from different user groups makes the development of enhanced access network necessary and the Cloud-RAN (C-RAN) is recently proposed widely for future cellular networks. The C-RAN is an umbrella term for a centralized, cloud computing based architecture for RANs that supports current as well as future wireless communication standards from typical radio access nodes to low-powered radio access nodes. The C-RANs promise to cope with the predicted mobile traffic volume increase by combining different RATs in heterogeneous networks and enable flexible handover choices for a mobile user to a more suited access network. However, the challenge with C-RANs is that there are more processes of transferring the connection from one RAN to another when the end-user device is moving around. These challenges include efficient decision making for connection transfer to a suitable RAT considering the scarce access resource and the demands of all connected users. In particular, the collection of information needed for the handover decision that provides centralized control of different network elements while maintaining cooperation and information exchange between them is still a key issue in C-RAN. Another challenge is to provide a common platform that supports service tailored mobility control and resource allocation in given hybrid radio environment, more specifically when both networks that are associated with different wireless access technologies and seamless mobility need to be provided [71].

#### **4.5 Intersystem internetworking**

Loose coupling and tight coupling are two basic interworking approaches which have been defined by ETSI for the integration of the different types of radio access technologies [72]. In loose coupling, RATs are independently deployed and data do not pass through the common core network. For example, in 3GPP EPS context data do not pass through EPC and with unified subscription and authentication management between integrating

RATs which allow their data to go directly to the internet without requiring for a link between their components and the 3GPP core network. In tight coupling architecture, the multiple RATs data pass through a common core network before going to the internet. There are several challenges associated with intersystem internetworking such as the provision of a common authentication, authorization, and accounting (AAA) scheme, an end-to-end Quality Of Service (QoS) and seamless handover etc. The QoS provided by the source and target access systems should be nearly identical to sustain the same communication experience and handover latency should be according to application needs etc. More specifically in the tight coupling, certain modifications of current wireless access networks are necessary for providing seamless service to mobile users to move from one RAT to another. To appreciate the challenges, 3GPP Technical Specification [8], a study on architecture for next generation networks with separation of control and data plane, specified in this context the architectural requirements, key issues and their solutions with functions descriptions. The 3GPP's CN exposes generic, IP-based interfaces towards the different non-3GPP access networks [43]. With implied pre-registration to 3GPP over non-3GPP access, heterogeneous traffic is enabled to route through the common core network and subsequently, the core network services are available to the registered users. Furthermore, evolutions of 3GPP core network also include network control entities for keeping user subscription information, determining the identity and privileges of a user and tracking the activities, enforcing charging and QoS policies during intersystem internetworking. Whether an external access network is trusted or untrusted is not defined by the main access network itself and by network operator who administrates the core network functionalities and takes decision based on the access network deployment scenarios and security features. The master network provides the indication of external IP access as trusted or untrusted to the connecting mobile user and requires appropriate attachment and data flow procedures with the external access network. To support interworking via an untrusted access a specific node evolved Packet Data Gateway (ePDG) is introduced that maintain an IPsec tunnel with access network's services. In addition, for the allocation of a remote IP address, ePDG supports the functionalities of routing of packets between the PGW and the user terminal as Mobile Access Gateway (MAG) if network-based mobility is used and as Local Mobility Anchor (LMA) to serve as local home agent.

#### **4.5.1 Access Network Discovery and Selection Function (ANDSF)**

ANDSF is a core network entity first introduced in release 8 for 3GPP compliant mobile networks [73]. The purpose of the ANDSF is to assist a UE to discover non-3GPP access networks and to provide the UE with rules policing the connection to these networks. The architecture for ANDSF contains a reference point S14 to the UE and supports both pull and push methods, i.e. the UE can either request the information or the ANDSF can initiate the data transfer to the UE. The information exchange between UE and ANDSF entity is based on the Open Mobile Alliance (OMA) Device Management [74] and uses the ANDSF Management Object (MO) specified in the 3GPP standard [30] to manage the inter system mobility policy and access network discovery information provided by the ANDSF. The gained information in form of MOs is stored in the UE. The ANDSF policies are mainly categorized as inter system mobility policy, access network discovery information and inter system routing policy. All three types of information provided by the ANDSF can have validity conditions, which indicate when provided policies or information are valid or not. These validity conditions can be associated for example with locations or time.

#### **4.5.2 Media Independent Handover**

Like 3GPP other standardizing organization have also defined solutions for intersystem mobility in heterogeneous networks. Media Independent Handover (MIH) services is a solution proposed by IEEE special working group and is defined in IEEE 802.21 [75]. The main objective of MIH services is to optimise intersystem handovers between 3GPP and non-3GPP communication networks. To achieve this, the mobile user and network infrastructure are used to get as much information as possible about the surrounding environment. These technology independent link-layer intelligence and network environment information are then provided to upper layers independent of the MU and communication technologies. To perform handover between heterogeneous networks under MIH services, a multi-model device is required that has at least two interfaces. The IEEE 802.21 standard also defined other network entities such as MIH Point of Service (PoS) entity which exchange messages with one or more MUs and MIH Point of Attachment (PoA) entity that may have two sub-types of Serving-PoA that is the MU's current data link layer connection and Candidate-PoAs that are potential PoAs to establish a data link layer connection. The Service Access Point (SAP) interface has been provided

between MIH functions and each of the communication technologies. The standard provides its functionalities through a) Media Independent Event Service (MIES) that propagate events from lower to upper layers, b) Command Service (MICS) that send commands across different layers to check the status of a link like signal-to-noise ratio (SNR) and Bit Error Rate (BER), and c) Information Service (MIIS) that allows the MU to gather information about access networks in its surroundings to help the network selection algorithm.

In order to benefit from these services, the users and underlying network entities need to subscribe to receive specific event notifications. These events may include particular state transitions or link layer updates, such as handover completion notification and/or reports of changes in link conditions that have exceeded a specific threshold. These notifications can include the forecast information. For example, a decrease of the signal strength in case of a wireless connection can predict that the link layer connection will be lost soon. In addition, these forecast notifications enable in-time handover decision and further optimise the handover process by improving the co-operation of the link layer, network and transport layers. The standard defined a generic mechanism to exchange information of potential access network technologies and are provided through MIIS Information Elements (IE). Unlike command service this information is normally of a static nature and can be classified in following three groups:

- The general information and access specific information gives input of the available 3GPP and non-3GPP networks within given area. It includes charging policy, link layer QoS, used spectrum bands, the maximum bit rate, the network operator and the roaming partners.
- Point of Attachment (PoA) specific information are network nodes which terminate the data link layer, such as a Wi-Fi access point or a WiMAX base station. The information elements related to this group provide information about the PoAs of the available networks including the channel range, the link layer address and the location of the PoA.
- Access network, service or vendor specific information elements provide an overview about the supported network and transport layer protocols and supported external data networks.

The MIH standard mechanisms speed up the intersystem handover decision making and improve the selection of the most appropriate access network.

However, the information gathering procedures are time-consuming and are not energy efficient. The approach adopted by MIH does not cover a complete definition of the handover policy and execution. More specifically, in the considered automotive scenarios, the main concern is the high mobility with application-oriented handover and session continuity requirements across multiple wireless technologies. To cope with these requirements, it is necessary to enhance MIH for the transmission of both user preferences and operator policies. In this regard, the IEEE 802.21 standard can be adopted for integrating into an architecture for global interworking and user specific mobility support [76].

Based on the architecture enhancements in 3GPP Technical Specification (TS) 23.402 [73] for non-3GPP accesses, in a related work [77] I presented prospects for TV white space access in the evolved packet core environment. As shown in Figure 4.4, the 3GPP-specific access technology represented by eNodeB is connected through the Serving Gateway (S-GW), the S-GW acts as the mobility anchor for mobility within 3GPP-specific access technologies and is directly connected to the PDN Gateway (PDN-GW).

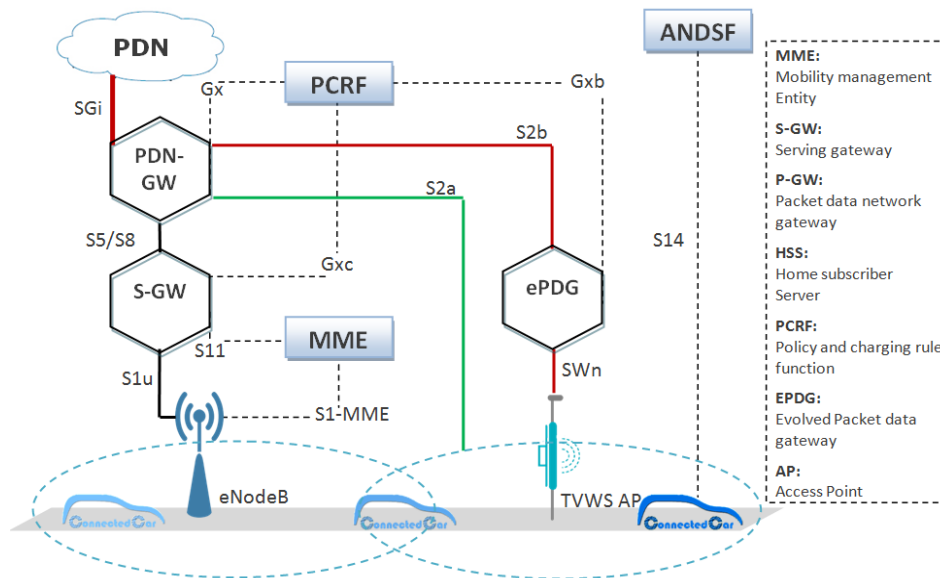


Figure 4.4: EPS architecture inter system interworking

The non-3GPP access network TVWS Access Point (TVWS AP) is also connected to PDN-GW when representing a trusted access network. In this approach, EPC is accessed through TVWS AP over S2a interface. As untrusted access network EPC is accessed through TVWS AP over SWn

interface using ePDG node. The ePDG acts as a termination node to established connection of MU (referred to as slave WSD using TVWS access terminology) as untrusted access to EPC. Slave WSD must also establish a secure IPSec Tunnel to ePDG node. All control messages and user data between EPC and slave device is then transferred through this tunnel. All data paths from the access networks are combined at the PGW. The presented architecture proposes Spectrum Coordinator (SC) as the means for cognitive communication between all system entities controls co-existing by managing and optimizing spectrum, specifically in inter system handover context SC and ANDSF cooperation in order to implement the specified method of operating with the White Space Database (WSDB) to reflect the operational constraints. The proposed cooperation also facilitates TVWS access policy information exchange among the different nodes and could use its enhanced context awareness to have target frequencies for intersystem handover identified in advance and therefore coordinate the handover more quickly than conventional radio access.

In line with considered use cases, in this thesis, the intersystem handover of a MU from a 3GPP access network towards a non 3GPP access network is described. I focus mainly on the operation of seamless connection mobility on network and transport layers with multiple PDN support approaches when applying SDN and virtualization technologies in mobile networks.

#### **4.6 Distributed Mobility Management (DMM)**

Existing IP mobility protocols rely on the use of a centralized and hierarchical architecture and several open issues have been identified by the research community [78]. With centralized mobile network architecture, since the connected MU uses the single IP address anchored at central IP anchor, during the handover user traffic will always need to go first to the home network and then to the correspondent service node even it is not connected via home access network, leading to paths that are in general longer than the direct one between the MU and its correspondent service node. This poses excessive traffic concentration on a single gateway element and possibly un-optimized routing adding in turn, unnecessary delay and resources usage. Centralized solutions are also probable to have a reliability problem, as the central IP anchor may potentially a single point of failure. One central IP anchor have to deal with higher user traffic



simultaneously, thus need to have enough processing and routing capabilities. This also implies certain scalability and network design challenges. To cope with problems of centralized mobility management: Distributed Mobility Management (DMM) has been introduced as a new paradigm to design flat and flexible mobility architecture. The basic concept of DMM is that a mobile unit no longer uses a single IP address anchored at

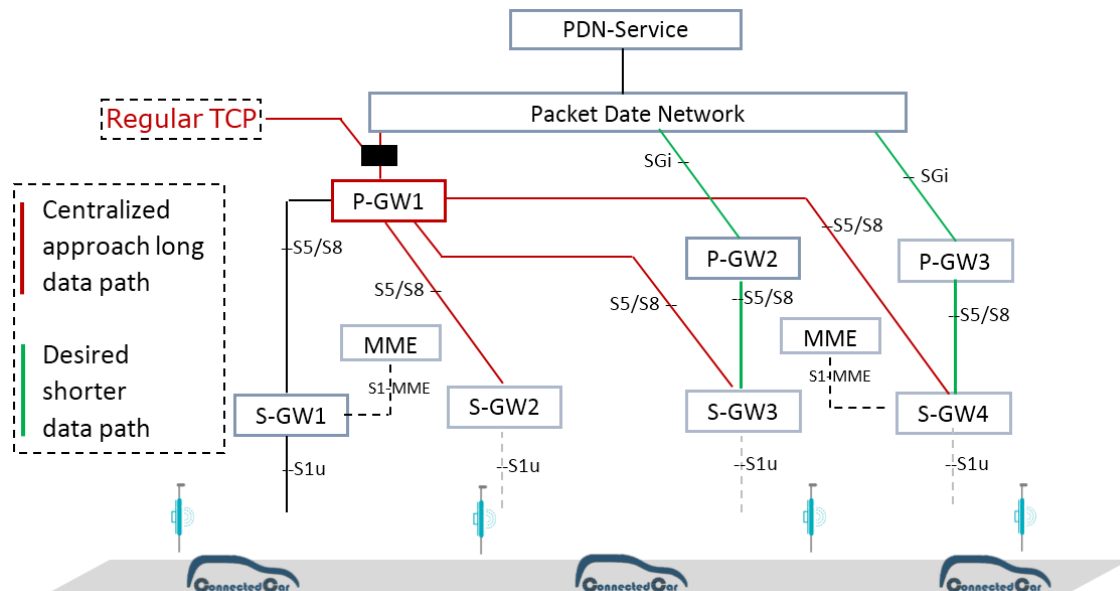


Figure 4.5: Baseline architecture for distributed mobility

a central home access network, but it deploys and uses a local IP anchor at visited access network to start new communications while maintaining the reachability of initial data connection that is still in use by active communications. More specifically DMM provides the distribution of mobility and IP anchors and selection and re-location of IP gateways when necessary that are topologically/geographically close to the MU. Figure 4.5 shows a reference architecture for DMM using 3GPP EPS terminologies. MU is initially connected to PDN-Service and during the mobility events a local S-GW can be used by a S-GW relocation procedure. This allows to select also a local P-GW and a new PDN connection is established with the desired shorter data path.

Furthermore, the implementation towards only packet-based solutions for both data as well as voice communications in current and upcoming mobile network architectures implies a more critical role for IP mobility management in providing the ubiquitous always-on network access service. In order to address the emerging needs of IP mobility, DMM extends and adapts already existing IP mobility protocols such as MIPv6, PMIPv6 and GTP to facilitate the networking architecture migration. In terms of solution

space, there are two main approaches for distributed GWs anchoring client-based approach and network-based approach.

In client-based mobility management approach (MIPv6), IP mobility is enabled by an entity called Home Agent (HA) which anchors the permanent IP address used by the MU, called the Home Address (HoA). As MU moves away from its home network, it acquires a temporal IP address from the visited network called Care of Address (CoA). The HA is responsible to maintain the MU's HoA and redirect traffic to and from the MU's current location. Following the proposal of distributing the anchoring IETF specified some extensions to MIPv6 [79]. As shown in Figure 4.6a, multiple HAs are deployed at the edge of the access network. The MU initially attaches to the distributed anchor HA/AR1 and configures the IPv6 address HoA1 to communicate with a correspondent PDN-Service. If MU moves to new HA/AR2, the MU has to keep bind home (while maintaining the reachability for those IP addresses that are still in use by active communications) address and configure the locally-anchored address to start new communications which is actually playing the role of care-of address in these bindings. IP mobility management is provided by the use of bi-directional tunnels between the MU and each one of the home agents anchoring in-use addresses. This approach requires additional intelligence on the mobile node side, as it has to manage multiple addresses simultaneously, select the right one to use for each communication, keep track of those addresses which need mobility support, and perform the required maintenance operations (i.e., binding signalling and tunnelling) [78].

With network-based approach such as in PMIPv6 as well as the GTP, IP mobility management is provided without the involvement of mobile users. Movement detection and signalling functionalities are performed through a network functional entity. Referred to as Mobile Access Gateway (MAG)/S-GW in IETF/3GPP context respectively. Figure 4.6b illustrates the operation of generic network-based DMM approaches. Mobility anchors are set up at the edge of the network thus anchoring and routing the local traffic for a given user. Furthermore, depending on the level of coupling of the control plane and the data plane, there are two sub-variants of network-based solutions fully and partially distributed. In a fully distributed model and using the PMIPv6 terminology each access router also behaves as local mobility anchor and implements both control and user plane functions.

In a partially distributed model, only the user plane is distributed and the control plane is separated and managed by a central control entity [78]. In this case, the operations are similar to 3GPP's EPS where the control plane is managed by the MME and the user plane by the S-GW and P-GW.



Fig. 4.6a: Client-based approach

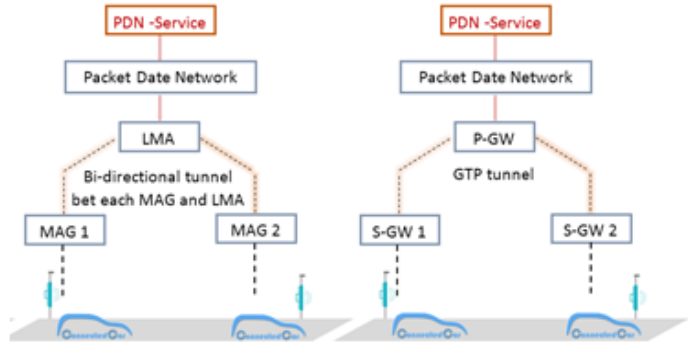


Figure 4.6b: Network-based approach

DMM approach solves some of the problems of centralized mobility management. however, when a MU moves to a new set of anchors, due to IP relocation, either tunnelling need be used between the initial router and new router or the active flows have to maintain until the flow is finished. Consequently, it may lead to a complex process and a high signalling cost. To cope with these challenges, in a related work [80], I presented improvements for distributed network connections to enable seamless IP mobility. The upper layer protocols based solution was proposed to remove the chains of IP preservation of current mobility management approaches, leading in turn, to seamlessly change the IP address during a connection without breaking and re-establishing the connection. I adopted the distributed co-located S-GW/P-GW for optimal data paths, the transport path length between the corresponding nodes has thus dwindled significantly in the architecture presented. I enhanced the proposed TCP protocol default scheduler to manage different types of IP flow. The pictorial call control flows thus created has shown a significant decrease in signalling and delay compared to tunnel and routing based approaches of distributed mobility management.

#### 4.7 Multiple PDN support

In order to support the simultaneous exchange of IP traffic to a single or multiple Packet Data Network (PDN) services through the use of multiple data plane IP anchors, there are several techniques under consideration within standardizing organizations and research communities, spanning

from enhancements of existing standardized protocols to clean-slate novel approaches. In line with considered cases for distribution of gateway functions, one of the solutions is the association of an active session with multiple prefixes based with modifications to classical IP mobility protocols, in particular of the well-known IPv6. In 3GPP recent technical specification [12] usage of an IPv6 multi-homing and usage of an UL Classifier are introduced. Both mechanisms are used to enable multi-homed PDU Session that provides access to the PDN service via more than one data plane IP anchors. The second category follows an ongoing effort within Internet Engineering Task Force (IETF) to support multipath operation, a set of extensions to enables a regular TCP connection to use multiple different IP addresses and interfaces [81]. In the following, I provide details of possible scenarios, a given MU to be connected to the PDN network over multiple network interfaces, each of which can be connected to an initial or to a new (local) data plane IP anchor.

#### 4.7.1 IPv6 multi-homing

In IPv6 multi-homing an additional UPF referred to as "Branching Point" is introduced in the data path of an active session, which provides forwarding of the UL traffic towards different data plane IP anchors and merges of DL traffic on a link towards the MU. As shown in Figure 4.7, the different data plane paths leading to the different UPFs branch out at "Branching Point", allows multiple PDU sessions to be built between a PDN service and a MU. Multi-homing of a PDU session supports only for PDU Sessions of IPv6 type. When the UE requests a PDU session of type "IPv4" it is provided with an indication to the UE that the network supports a Multi-homed IPv6 PDU session. The insertion and removal of a Branching Point is then decided by

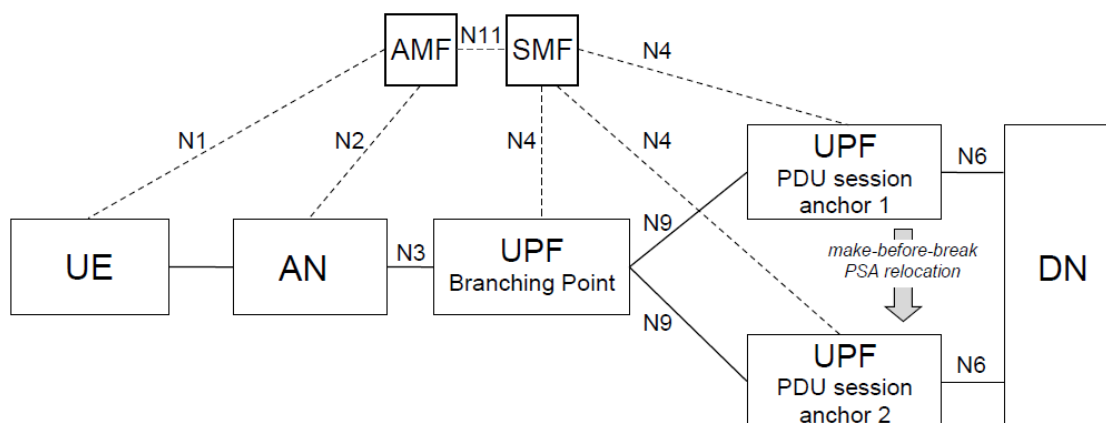


Figure 4.7: IPv6 multi-homing: overview and operations [12]

the SMF using generic UPF capabilities. The UPF applied as BP may also be controlled by the SMF to enforce required QoS parameters such as traffic measurement for charging, traffic replication for physical layer and bit rate enforcement per session etc. The SMF can decide to insert a UPF supporting Branching Point functionality in the data path of a PDU session during or after the given session establishment or to remove from the data path of a PDU Session after the session establishment.

#### **4.7.2 Multipath TCP (MPTCP)**

Multipath TCP is newly proposed extension to TCP that enables the use of multiple addresses/interfaces for the transmission of data packets that belong to a single connection and to be sent from different interfaces, without requiring any changes of the applications [82]. MPTCP does not change the socket API and it can be used transparently by any data network, allowing them to seamlessly move data connection from one data path to others. For this purpose, each MPTCP connection is branched into several TCP connections called subflows. To create a connection with a PDN service, an MPTCP enabled MU with cellular and Wi-Fi interface sends a SYN segment over the initial (for example over a Wi-Fi) interface. This segment includes a random key and contains the MP\_CAPABLE option to indicate that MU is MPTCP capable. The PDN service acknowledges with a SYN+ACK segment that also contains a random key and MP\_CAPABLE option. The MPTCP connection is established over the Wi-Fi interface by the three-way handshake. At this point, the connection is composed of only one subflow and the data can be sent over this initial subflow. MPTCP uses two levels of sequence numbers, the regular sequence number in the TCP header that anchor the bytes sent over this specific subflow and the data sequence number that is placed in the Data Sequence Signal (DSS) option and tracks the bytes carried over the entire MPTCP connection. To use the second interface, the MU sends a SYN segment with the MP\_JOIN option over the second (in this case a cellular) interface to use both interfaces. This option includes a token derived from the random key exchanged in the initial MP\_CAPABLE option segment to identify the MPTCP connection to which the subflow must be associated. The connected PDN service acknowledges the second subflow establishment with a SYN+ACK segment containing the MP\_JOIN option. The MU finalises the three-way handshake with an ACK segment and the new subflow is established. At this point, the MU and the connected PDN service can send data over the WiFi and/or the cellular

interface. Multipath TCP ADD\_ADDR option and REMOVE\_ADDR option [81] can be used to add or remove subflows at any point of time in any ongoing MPTCP communication. In this way, the change of IP address of a MU does not force the connection to be restarted and could be used to remove the chains of IP preservation of current mobility management solutions and disburden the process of flow forwarding during data plane IP anchor relocation.

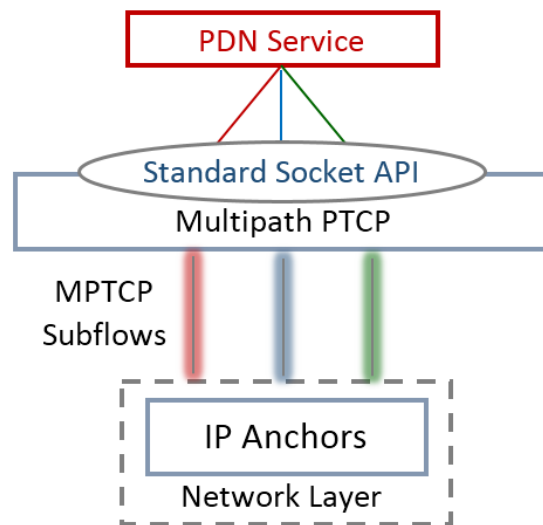


Figure 4.8: Multipath TCP multipath operations

The Multipath TCP implementation uses a packet scheduler to decide over which available subflow data to be transmitted. The scheduler has access to the state of each subflow, including congestion window (cwnd) and the order of time-distance estimation i.e., Round Trip Time (RTT). The scheduler identified multiple available paths by the path index and based on applied scheme (i.e., largest bandwidth or lowest packet delivery delay etc.) packets are scheduled to available subflows. The current Multipath Linux kernel implementation contains two basic strategies that are referred as full-mesh and ndiffports path managers [83] to perform management of the subflows. The full-mesh path manager establishes one subflow over each active interface. In this case, path manager listens to events from the underlying network interfaces and establishes subflows immediately when an interface becomes active. In case of ndiffports path management, instead of using multiple IP addresses this path manager always uses the same IP address pair for its paths. Immediately after the establishment of

the connection ndiffports path manager creates n subflows over the same interface as the initial one. Most relevant use case for ndiffports path management is datacentre redundant network where it enables the utilisation of paths that are load-balanced with Equal Cost Multipath (ECMP).

#### 4.8 Important performance metrics

I consider the following key technical performance parameters that an optimal automotive access system is expected to provide. It also provides the necessary related information about the individual performance metric and values chosen. The intent is to define the conditions against which the performance of the proposed solutions will be compared.

<b>Performance Metric</b>	<b>DESCRIPTION</b>
Area traffic capacity	This is the total available traffic capacity per geographic area, i.e. the number of correctly received Mbit/s/m <sup>2</sup> over a certain time period or the number of bits contained in the service data units delivered to the network layer in a given time. For automotive access, this can be derived from network deployment (e.g., site density), average spectral efficiency and bandwidth. In the case of aggregated bandwidth across multiple bands, the area traffic capacity will be a total of the used bands. The target value and conditions including supportable bandwidth as described in International Telecommunication Union (ITU) report for area traffic capacity in downlink is 10 Mbit/s [48] for the purpose of evaluation in the automotive internet test environment.
Connection density	It is the total number of mobile devices fulfilling a specific QoS per unit area (in an automotive environment per square kilometre).
<b>Quality of Service</b>	
Allocating priorities	Allocating priorities to different application flows to offer a specific level of network performance for the respective traffic. Efficient allocation of the opportunistic resources should fulfil the QoS needs inside given bounds specific to the scenario and during location update sustain the same communication experience provided by the source and target access systems. To support a specific QoS, the delivery of a message of a certain size within a certain time and with a certain success probability need to be provided. For example, for brake controllers in platooning, a very low packet

	error rate (e.g. less than $10^{-6}$ ) are desirable [68]. Furthermore, the allocation to be made dynamic with policies based on the connected car's perspectives including but not limited to the QoS, the message priority and security.
<b>Mobility Management</b>	
Mobility models	Two main classes for automotive access are specified in [67] with speed from 120Km/h to 500 km/h for high speed vehicles and from 10 km/h to 120 km/h other vehicles. The regular updates and consult to relevant nodes are needed accordingly to the mobility and speed of connected vehicles.
Handover execution time	An important parameter in automotive access is the time taken to execute a location update procedure which includes the time elapsed since the old access link is torn down until the new one is established. The Handover execution time is the sum of link, network and transport layers handover times and the handover interruption time.
Mobility interruption time	Mobility interruption time is the shortest time duration supported by the system during which a mobile unit cannot exchange data plane packets with any access node during transitions. This includes the time required to execute any access network procedure, radio resource control signalling protocol, or other message exchanges between the vehicles and the access network. The requirement defined for mobility interruption time is 1 ms for the purpose of evaluation in the automotive URLLC scenarios [84].
Average latency of data packet delivery	It represents the downlink path average delay (ms) of the first data packet received by the moving vehicle after handover execution. This is the mean of the contributions of data plane and control plane delays. The data plane latency is when the source sends a packet to when the destination receives it. It is defined as the one-way time it takes to successfully deliver an application layer packet/message from the radio protocol layers ingress point to the radio protocol layers service data unit egress point of the radio interface in either uplink or downlink in the network for a given service in unloaded conditions, assuming the vehicle is in connected active state. The inter vehicle cooperative awareness message poses the maximum tolerable end to end latency of approximately 3.3 ms [68] for traffic



	safety applications, 4 ms for infotainment application and 1 ms for URLL automotive application.
Cumulative Distribution Function (CDF)	CDF of the latency is used to observe the maximum latency of the first optimal path redirected data packet after the completion of the handover procedure. The CDF will be used to demonstrate how the measured redirected data packets latencies fit with respect to the standardised threshold.
<b>Traffic Steering</b>	
Admission Control	The admission control should depend on the QoS requirements and on the channel capacity of the available resources. The required level of QoS should be maintained in the presence of variations in the available spectrum resources.
Specific to Scenario	

## 4.9 Key enabling technologies

The following sub-sections briefly highlight key emerging technologies in centralizing network control, introducing network programmability and sharing the network fabric that could shape the design of future mobile networks.

### 4.9.1 Software Defined Networking (SDN)

SDN features various ways of using software engineering technologies to manage and manipulate network devices in which the control of gateway functions are lifted up to a logically centralized entity called SDN controller. The key concepts of SDN include the separation of the control plane and data plane, flexibility, high rate of innovation, and network programmability that speeds the deployment of new services. Figure 4.9 illustrates a typical architecture of an open SDN model. Data plane comprises a set of forwarding devices, (e.g., virtual switches or physical switches). The control plane consists of SDN controllers that provide the consolidated control and supervision functionalities and manage the network forwarding behaviour through an (open) interface. The application plane consists of one or more business applications, such as routing, and monitoring applications. The applications consume the SDN communication services and communicate their network requirements towards the controllers via northbound interfaces. The different controllers communicate with each other using east/westbound interfaces. The boundary between the data plane and the control plane is traversed through the southbound interface to configure,

monitor and manage forwarding devices. For example, when a data flow arrives at ingress switch and does not match any rules in the flow table, the first data packet of the flow is sent to the SDN controller. The SDN controller by interacting with related SDN application computed the forwarding path for the flow and sends the appropriate forwarding entries to install in the flow tables at each switch of the defined data path. All subsequent data packets of this flow and different flows with matching rules in the flow table are forwarded to the defined data path.

OpenFlow protocol [85] is the most extended SDN interface maintained by the Open Networking Foundation (ONF). The ONF has established the Wireless and Mobile Working Group [86] to explore how the OpenFlow can adapt for use in mobile networks. In a solution brief [87] they illustrate the benefits of SDN/OpenFlow with use cases of inter-cell interference management, Mobile traffic management and envisioned that the SDN/OpenFlow benefits can also be realized throughout the mobile networks from access to backhaul and to the EPC. Open SDN makes it easier

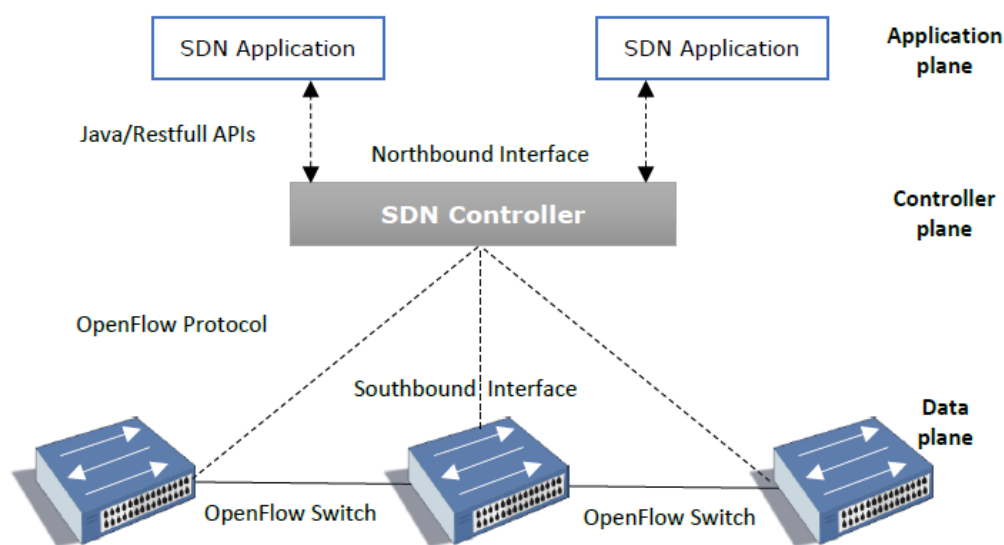


Figure 4.9: Typical Open SDN Architecture

to introduce and deploy new applications and services. However, the effectiveness and great potential of SDN for mobile networking are faced with new challenges which need to be addressed by the new research advances [88].

#### 4.9.2 Virtualization

Virtualization is a useful technology that allows multiple users to share the same physical infrastructure by abstracting and decoupling the computing

functionalities from the underlying hardware. Similarly, in considered context, Network Function Virtualization (NFV) also known as network function cloudification, enables the virtualization of entire network functions that are tied to fixed configuration of hardware in conventional architectures, causing operational inflexibility. ETSI's architectural NFV options are described in [89], which consists of four main elements namely NFV Infrastructure (NFVI), Virtual Network Functions (VNFs), hypervisors, and NFV Management and Orchestration (NFV MANO). The VNFs is the key component of this architecture that are the software implementations of network functions and are deployed upon a generic cloud infrastructure. During recent years virtualization technologies have moved from server and network domains to virtualization of mobile network functionalities. New applications of network virtualization can enable more flexible management of network interconnections and can potentially enable different resources to coexist on the same virtualized infrastructure [90]. These resources can be any network components, such as switches, routers, RRHs, storage servers, physical or logical (soft) connection between two nodes in the network, virtual machines etc. Some interfacing resources, such as network interfaces, can be considered a part of both host/server and network virtualization. These virtual resources, created by hypervisors, bring virtualization over physical hardware. Virtual instances of mobile Network are the collection of logical components of different radio access and network nodes. These are directly interacting with the physical resources. Although virtual instances are only logically isolated networks over a common infrastructure, but each such virtual network can provide the user of logical virtual networks, network services similar to those provided by the common infrastructure as a non-virtualized network. There are different scopes and perspectives of mobile networks virtualization that are suitable for different applications and services that can be combined with configurable SDN technologies to address a number of challenges in the current networks.

Both SDN and virtualization have related mechanisms and therefore could complement one another. Control and management for computing resources based on virtualization and software defined technologies have been extensively studied and for cloud computing application is being tested in major field trials. Using similar logic, recently, many research works have tried to redesign the traditional mobile network using two of these concepts in order to deal with the challenges faced by mobile operators, such as the

rapid growth of mobile traffic and new services. As the mobile medium is an ever-changing environment, there is a need for the virtualization techniques used in host virtualization to be modified or adapted when applied to mobile network technologies.

In addition to the technologies discussed above, I cite the following technologies in the evolution path that will also impact the design and development of future mobile networks.

Machine Learning (ML) [91] and Artificial Intelligence (AI) [92] technologies can enable future mobile networks to learn from the experience and modify itself and accommodate new services. For example, AI technology can observe the signal strength, learn patterns and predict handover events. The precious radio spectrum in the lower bands will be used more effectively by allocating the frequencies every second or so, based on the context. Multi-tenancy will further be enhanced and will allow each virtualized network to have a distinct policy irrespective of being controlled by an infrastructure provider, single enterprise network, virtual mobile operator, government administration service network, or other operating bodies. The massive use of multiple antenna systems will be able to exploit the multi-rays propagation, for tera hertz throughput but also for precise localization and for energy transfer. And there are some other novel techniques that have very promising features and related challenges, but to limit the potentially huge design space and envisioned research I focus mainly on SDN and virtualization technologies.

#### **4.9.3 Mobile Network Slicing**

The Network Slicing is the most relevant feature of 5G Systems to serve vertical industries with widely different service needs. It does so by exposing isolated partitions of network resources and services for optimized wireless communications, e.g. in the areas of domain specific reliability, capacity, latency and isolation [20]. A network slice is a complete logical network including R(AN) and CN resources that provide certain network capabilities, supported features and network functions optimizations. Emerging techniques like virtualization and SDN are used for network slicing. As described earlier in section 4.9.2 virtualization allows to share the physical infrastructure by abstracting and separating the computing functionalities from underlying hardware [90]. New appellations of network

virtualization can enable more flexible management of network interconnections, resource abstraction, resource partitioning, resource aggregation, centralization of control, and can potentially enable different resources to coexist on the same virtualized infrastructure. These resources can be any network components, such as switches, routers, a physical or logical (soft) connection between two nodes in the network, virtual machines etc. Some interfacing resources, such as network interfaces, can be considered a part of both host/server and network virtualization. The collection of logical components of different radio access and network nodes (Virtual instances) of a mobile network are directly interacting with the physical resources. Although virtual instances are only logically isolated networks over a common infrastructure, but each such virtual network can provide the user of logical virtual networks, network services similar to those provided by the common infrastructure as a non virtualized network.

To this effect, during recent years network slicing has also attracted a lot of research interest in academia and standardizing organizations. In [93] the authors proposed that network slicing mechanism for network edge nodes, where the centralized core network entities and associated applications are shifted to the network edge to offer low latency services to mobile users and reduce burdens on the backhaul. The authors of [93] also proposed network slicing based mobility management schemes and an optimal gateway selection algorithm was devised to support seamless handover. In [94] an algorithm was proposed based on SDN to optimize the virtualized radio resource management with resource pooling, a user oriented service slicing strategy that considers different QoS requirements. The authors in [95] introduced a scheme for resource allocation and interference management. They also validated that the proposed scheme guaranteed the different QoS requirements by optimizing power and subchannel allocation jointly. In [96] an architecture based on SDN and agile technologies was presented that enables allocation of the physical network resource to virtual slices and perform scheduling among them. The work closest to this work is proposed in [84], there the authors elaborated the technological options, enablers and concerns for the successful implementation of 5G slices for multi-tenant V2X communication services.

#### 4.10 **Final remarks. The need for customised vehicular access**

The main challenge comes of today's mobile networks is that multiple services are supported over the same architecture, which would not natively support specified V2X communications and typically conceived with no elasticity in mind and are processed by the same network elements in the core network and by sharing the same resources in the access network. For example, the cooperative ITS messages dissemination over a given access coverage area may reach vehicles that are not interested (e.g., cars moving in the opposite direction to the advertised road hazard notifications, or vehicles outside the cooperative awareness range). Therefore, specialized network entities and other core network elements should be involved and designed for vehicular messages broadcasting.

In response to these observations, in early attempts to adequate V2V use cases support, new features of the current status of technology for cellular direct communication, building on the Device-to-Device (D2D) communication protocol was identified as part of 3GPP recent releases. This enables device relaying, allowing devices in a network to function as transmission relays for each other and realize a large-scale ad-hoc mesh network. Moreover, 3GPP is defining the full set of V2X technical enablers from releases 14 and 15 and, customized network approach was conceived to deploy physical infrastructure for each service (or even one for each business) with multiple Dedicated Core networks (DECOR) running on purpose-built proprietary hardware hosting the related network functions to support different services. This approach clearly cannot be applied in an effective way and calls for technical solutions that allow for both efficient resource sharing and multi-tenant infrastructure utilization. In this context, 5G Systems come up with several evolutions to provide customized networks and provide optimized wireless communications for a specific user group such as network slicing, proximity services, and mobile edge computing etc. I explore the technical details of these building blocks in the view of assessing for automotive access. Proximity Services not only provides a platform for the most desirable safety vehicular communications but also paves the way toward determining the source of autonomous vehicle attacks. Mobile edge computing promises to reduce latencies for some vehicular applications such as the traffic information system that have flexible latency requirements. Network slicing has the capability to

facilitates dynamic and efficient allocation of network resources and support diverse service scenarios and services. There is a general understanding these enhancements will address some of the deficiencies of present vehicular communications standards. Henceforth, in this work, I explore the network slicing framework for vehicular communication and design a slicing scheme to be generic and adaptable to the specific user class.

## 5 S Plus Mobile Network

This chapter presents the major design decisions of the thesis, proposed automotive cognitive access and highlights some of its promising features. It begins with an overview of designed model of Software defined plus virtualization featured mobile network (S+ MN) architecture and description of its elements (refer to section 5.1). Sections 5.2 and 5.3 describes the enhanced behaviour of S+ MN architecture for the collection of network resources and create customized network instance to provide optimized wireless communications for a specific user group. The next sections present the automotive slicing and specific issues need to be addressed while applying the proposed slicing scheme for vehicular communications. The last sections describe a specifically configured automotive slice instance and discuss some exemplary procedures to highlight the key features of proposed automotive slicing architecture.

### 5.1 S+ MN Architecture Overview

To fully appreciate the particular challenges defined in section 3 and in the line with design selections described in section 4, I set out to take up the current mobile network architecture and designed an architecture named Software defined plus virtualization featured Mobile Network (S+ MN).

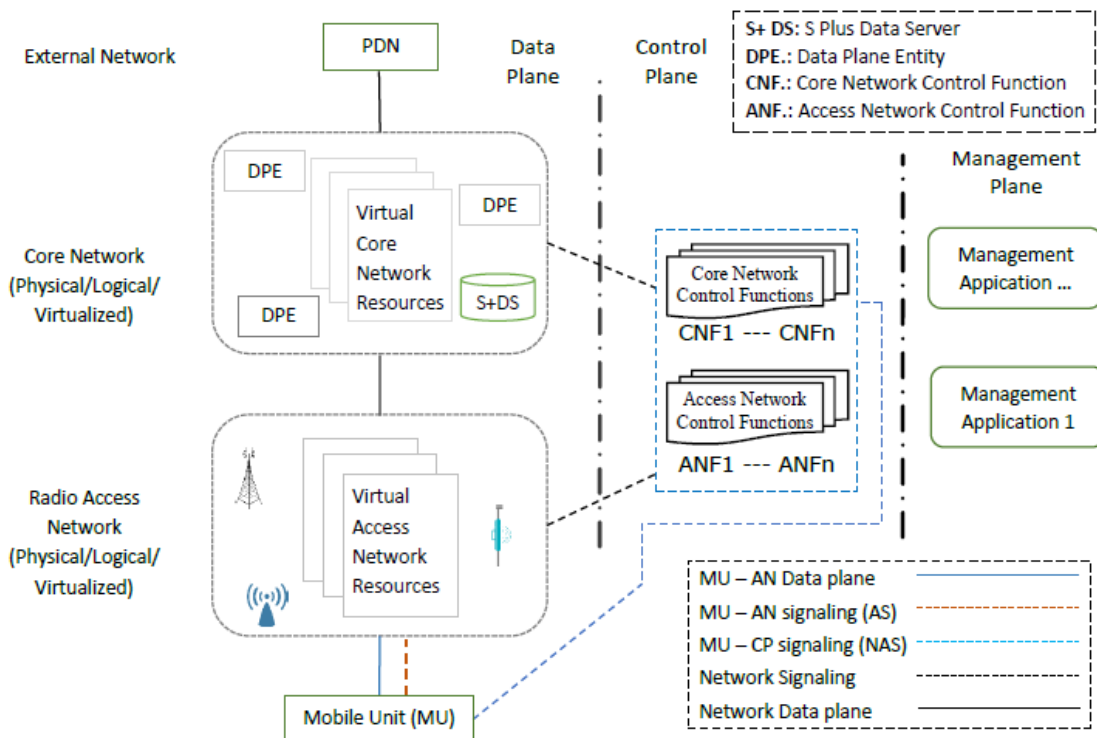


Figure 5.1: Generic view of the S+ MN architecture



Figure 5.1 shows a generic view of the proposed S+ MN architecture that represents the following key principles of re-architecting the current mobile network architecture:

- a) The control plane functions are decoupled from the data plane functions allowing independent scalability and evolution. The control path is extended down to the access and user nodes to enable independent access and data session handling.
- b) Allow access via multiple heterogeneous access networks and expose generic, IP based interfaces towards the different non 3GPP access networks.
- c) Distribution of gateway functions is followed and (Re)selection of efficient data plane paths is enabled with implied upper layer service continuity mechanisms to remove the chains of IP address preservation for session continuity during IP anchor relocation.
- d) Various types of user centric, network centric and context centric data is collected and managed in unified data server and use the additional information from big data analytics within the system to optimize QoS and Mobility control and management.
- e) The physical resources can be virtualized or logically partitioned from underlying hardware resources, thus has the capability to enable end to end network sharing traversing both (R)AN and CN.
- f) Allow to create a customised network to provide an optimized solution for a certain user class that demands specific treatment, e.g. in terms of extended coverage, seamless mobility, isolation, ultra-low latency and very high reliability. The QoS control and management will be flow based that enforce the required QoS over both the wireless access and the forwarding path.

These principles are applied to each level of S+ MN architecture and are used to provide a baseline model where this research is placed.

To fully appreciate the particular challenges described in section 4.11, I next detail the enhancements of the S+ MN architecture for the collection of network resources and create customized networks to provide optimized wireless communications for certain kind of communications.

## 5.2 S+ MN Slicing scheme

Advocated with ongoing standardization and research activities, network slicing is being established as the most capable framework to overturn the traditional approaches for resource sharing, consolidating resources into a single operation and multi-tenant infrastructure utilization. The slicing based mobile communication system is envisioned to provide customized network services for diverse user groups. However, the use of network slicing in future mobile systems opens up new concerns and are used to describe as follows. Most of the challenges arise from its limited specifications of its procedures. For instance, to give a truly modular approach for different sliced networks, different application requirements must be carefully translated and categorized into technical specifications. This will ensure that one sliced network does not affect another, and changes from one slice to another are seamlessly integrated [97]. Another aspect is that the classification of user requirements in order to determine if the network functions need to be centralized or sliced has not been sufficiently studied in slicing-based systems. Furthermore, the virtualized resource allocation and to be useful for decision making, new data analytics techniques must be developed for the dynamic network environment [97]. Furthermore, the emerging 5G services exhibit different demands for mobility control and management e.g., in the area of mobile unit speed that requires efficient tracking area update procedures and mechanisms to connect with multiple different access networks with efficient handoff schemes. The mobility management scheme for specially configured slice needs to be selected flexibly according to the context of the underline application. In particular, how does the path optimization work between MU and Application Function in a fixed data network using upper layer service continuity mechanisms (multi-homing/multipath) and how to keep the IP session continuity during slice switching. Moreover, in the roaming case, where a visiting network slice is not supported, how to manage and support such network slice switching needs further discussion. To obtain user state from one virtual entity to the other or preempt the user state handling priority may require extra procedure in the sliced core network. Finally, a consistent QoS control and management framework that enables optimal service level quality as per application needs, optimizing network capacity utilization during slice switching.

Having provided the variety of future wireless access services and their specific requirements, an SDN and NFV enabled mobile network slicing scheme is designed to be generic and adaptable to the specific user class. As depicted in Figure 5.2, it comprises three main components: the physical resources layer, the Slice orchestration layer (SOL) and the Slice Instance layer (SIL). The physical layer refers to the S+ MN System common resources deployed in a network that supports the procedures, information and configurations specified for network slicing. The Access Network (AN) comprises the heterogeneous radio access environment that includes legacy and novel 3GPP (like E-UTRAN, 5G New Radio) and also non-3GPP (like WLAN) access technologies. The physical RAN resources can be abstracted and sliced into virtual R(AN) resources which directly interact with the physical radio resources and collectively refereed as an Access Unit (S+ AU). The wireless connection between the radio access node and the MU shall be managed by the access control. The S+ MN Architecture enhances standard access node by adding customised functions for admission control, radio resource scheduling, local location management and inter access nodes routing, backhaul links management and QoS etc. More details of respective roles of each of these functionalities are described in later sections of exemplary procedures. The network control information is delivered from access control to access nodes through the standardized interfaces and apply access network-wide functions.

The data plane is an open, programmable, and virtualizable network forwarding infrastructure, which consists of evolved S+ access nodes and S+ Data Plane Entities (DPE), explicitly separate from the control plane. DPEs can be implemented on software based virtual switches, which supports various data plane operations. such as packet routing and forwarding operations, service detection operations, packet inspection, QoS handling for data plane, policy rule enforcement, bitrate enforcement operations, the anchor point for Intra/Inter RAT mobility (when applicable), external PDN session point of interconnect, traffic accounting and reporting etc.). Mobile edge computing platforms can also be used at the access level which can collaboratively execute the transmission of data in an efficient way. Based on the actual 3GPP specifications [12], [46] the flow-based QoS concept is followed, the data plane function inserts QoS marking (a scalar value) in the packets and packet flows of the same QoS marking belong to the same QoS flow. In order to apply the correct QoS parameters to the

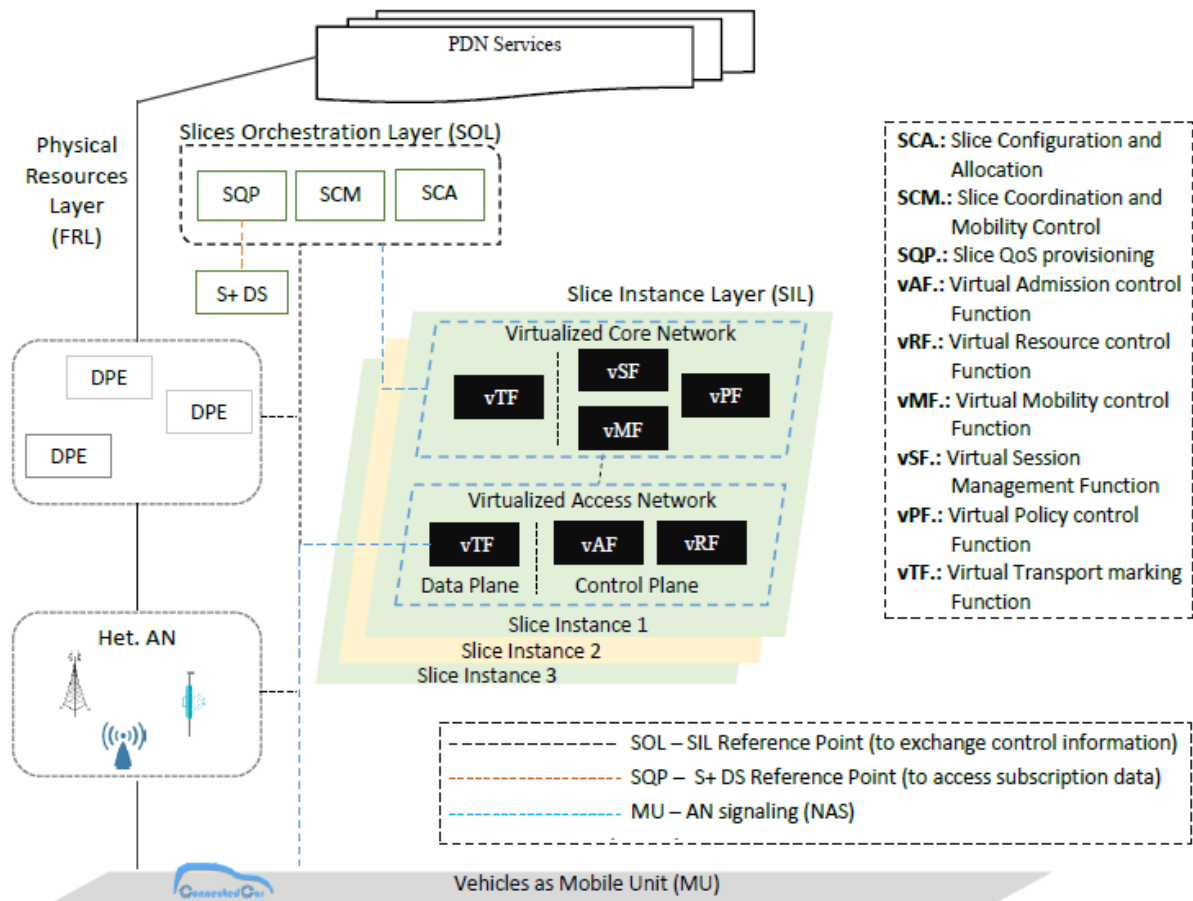


Figure 5.2: S+ MN Slicing Architecture

application's service data flows, S+ MN needs to know the application requirements, common control information as well as the user and service individual control information. To this end, the legacy subscription and policy entities are evolved to a unified shared Data Server (S+ DS). The S+ DS stores user subscription data and session related context (e.g. QoS, charging and user location etc.). The common control information is extended with individual control information and policies per user. The S+ DS maintains the list of services provided by the network. The architecture allows the S+ DS to collect various types of user-centric, network-centric and context-centric data. The real time aggregated data can be used in various forms. For example, if a certain data set is changed or threshold is exceeded. The context aware provisioning service can subscribe to certain events at the S+ DS and be notified by the S+ DS. As a result of such a notification through the service interface, the S+ context acquisition can load the current information into the S+ DB. This further enables real time insights for network planning, efficient resource management and later

apply to QoS and mobility control and management (e.g. offloading decisions, traffic routing and context aware service provisioning. etc.).

In slice orchestration, the control applications interact with each other through the interface protocols and provide the configuration, allocation, coordination, mobility and QoS provisioning functions of the network slice instances. Using a software defined approach, the slice orchestration applies network-wide functions across different slices, such as resource control support, virtual forwarders and service functions, inter slice mobility management, load balancing etc. The choices of orchestration control applications are guided by the tenants through Service Level Agreements (SLAs) and by the measured feedback statistics of physical resources.

The Slice Configuration and Allocation (SCA) is the core part of slice orchestration, which first perform the collection of the necessary information about vertical segments specific capabilities, usage type, mobility and service characteristics from subscription database and service requirements. The SCA then translate necessary information to resource requirements and identifying different virtual functions required to support the given services and generate slice context for service, control, and data plane management. The SCA then instantiate an appropriate slice instance based on the given information. It also includes the identification of interfaces, determining the allowed slice types, the parameter settings in the R(AN) and select corresponding dedicated control functions in such a way that it can operate alone for a specific reservation period in a specific area. Slice allocation applies mapping of physical network functions and allocates virtualized resources based on the given slice configuration. It is also responsible for determining the access and mobility functional grouping serving the MU and triggers the slice instance selection during user registration procedure.

Another main functionality within the orchestration layer is slice QoS Provisioning (SQP), By utilizing SDN capabilities the SQP application interfaces with the domain controller and the slice instance virtual functions distributed in the AN and CN. The SQM ensure the QoS provisioning according to service requirements by monitoring their status, adapt and maintain the slice instance in case of context changes, dynamics of networks, and location updates of mobile units. This may also include the

handling of configured vNFs life cycle management, failure management, and network reliability based on the Service Layer Agreement (SLA) requirements. Moreover, this enables inter slice internetworking for load balancing by monitoring their resource utilization.

Finally, the Slice Coordination and Mobility (SCM) control functionality assures the proper operation between slice instances and takes decisions based on the policies provided by S+ DB as well as supplemental resource usage data from different slice instances using its service. In automotive slicing context SCA forwards mobility requirements to selected slice instance: e.g. mapping mobility pattern, coverage requirements patterns, TAU hierarchy dependent on speed. The SCM is not specific to a particular slice instance functionality and is logically part of all slice instances.

The Slice Instance Layer (SIL) that includes slice specific functions which are deployed as virtualized network functions, comprises the collection of logical components of S+ MN and directly interacts with the physical resources. The slice instance includes virtualized slice Control Plane Network Functions (vCPF) and Data Plane Network Functions (vDPF) that are not shared with other network slices. Each slice instance maintains specific configurations of network functions according to the related context of the network slicing scheme. Multiple slice instances can deliver exactly the same features but for different user groups and may differ for network functions optimizations concurrently supported for a MU. The virtualized functions can be added or removed for certain use case requirement. The specific roles of each of these functions are described in section 5.5.

### **5.3 Automotive Slicing**

The proposed slicing scheme leads to the deployment of multiple slice instances based on the different use cases operational needs. Aiming at to design an automotive slice that can provide optimized access services for given use cases, following specific issues that needs to be addressed while applying the proposed slicing scheme for vehicular communications:

- **Characterisation of Automotive Slice Instance:** An important task for automotive slicing is the characterization of each slice instance to vehicular application context and the context information should be

efficiently distributed in the network in order to provide fast lookup procedures of the best slice instance for application requirements.

- Identification and rating automotive context information: Vehicular communications require customized paging and tracking area update procedures in order to support efficient automotive location services whilst keeping signalling traffic under control. This is specified in [8] as multi-dimensional slice descriptor and includes other parameters (e.g., the vehicle manufacturer, the road authority) that better identify the slice configuration, besides the connecting vehicle tenant ID and different slice instance reference types.
- Creating an appropriate number of slice instance: For verity of different use cases of vehicular communications [84] as described earlier in section 4.2, it's not appropriate either mapping into a single automotive slice or straightforward mapping into reference slice types. In addition, the granularity of controlling and managing the slices in automotive context (per flow, per service type, per device type, etc.) also need further optimization.
- Context Modeling and Prediction: Introducing integration of context prediction in addition to context awareness allows to start the related management tasks pro-actively in advance. For example, mobility prediction can be used to allocate an automotive slice instance based on a detected movement pattern, and accordingly be used to initiate slice configuration in advance before the vehicle moves to the next slice. Furthermore, context modelling may also include geographical position parameters [8] to allocate different resource pools to connected vehicles moving in opposite lanes.
- Vehicles would likely be conceived as multi slice mobile units, this implied them to be able to attach to multiple slices. Multi-tenancy is a typical feature of the possibly simultaneously offered V2X services in different scenarios. An autonomous vehicle driver could start his self-driving vehicle that need to exchange very low latency V2V messages that is offered by an autonomous driving slice, whereas the children sitting in the back seats use streaming of a video provided as a vehicular internet and infotainment slice instance, and a V2X service owned by given car manufacturer running in the background for remote diagnostic purposes provided is provided by related slice instance. To enable fast lookup procedures of the best slice instance for given scenario, the distributed context information in the network

are used to maintain and adapt the automotive slice instance in presence of context changes, dynamics of networks.

- The capability of providing efficient mobility support to connected vehicles as well as their service flows is predominantly seen as a challenge in terms of changing network (virtual and physical) conditions. The configuration of automotive slice instance and mapping of the mobility management scheme needs to be done based on predefined policies which are provided by the policy function and real time context information. Moreover, the vehicles require more handovers in the same or in different registration areas. Specialized handoff policies and settings at the access and core network level, efficient mobility anchoring with multipath support during gateway relocation and adaptive handover thresholds should be exploited. This also includes forwarding the session requirements to the corresponding session management function in handover scenarios, and specifically their collaboration to implement the application oriented forwarding of IP traffic.
- In addition to mobility, QoS should also be improved, allowing a per slice configuration tailored for application's specific QoS demands. Given the different set of QoS characteristics by the diversity of the V2X applications and traffic, in conjunction with their increasing expectations, the task of QoS provisioning needs to be approached from the perspective of providing priority to specific traffic types over coexisting services either through explicit resource reservation, or traffic classification using static policies. QoS aware routing with low delays for connection mobility is predominantly seen as a challenge in slicing-based systems.

In order to embrace these directions, Specific functional groupings need to be defined and each encompasses a number of individual functions for customized solutions for certain operational and service requirements.

#### **5.4 Automotive Slice Instance**

Figure 5.3 illustrates a specifically configured Automotive Slice Instance (ASI), a complete instantiated logical network including access and core network functionalities. To enable independent access and data session handling and following mobile edge computing features, automotive slice



instance is comprised by two functional levels in charge of respective duties at the R(AN) and CN domains. The network functions are elaborated by separation of data and control plane. At the access level there are three sub groupings of control functions i) ASI access and mobility control, ii) ASI Resource Control and, iii) ASI access level QoS.

ASI access and mobility control functional grouping includes automotive context acquisition, a block for the gathering of information on the environment and link level handover management. These functions can interact with subscription and content server S+ DS to create a local copy

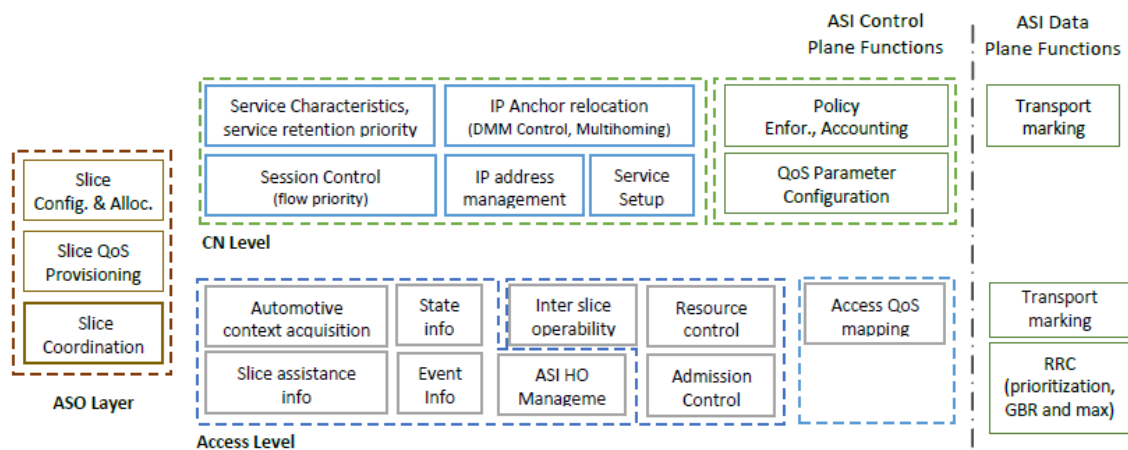


Figure 5.3: Automotive Slice (potential) functional grouping

for slice specific MU authentication and authorization. For mobility management state info and event info sub-functions tracks and updates the MU's location information as well as their requirements. ASI access and mobility control functional grouping also delegates various event to S+ access nodes such as initial context setup request and response for initial attachment. It also translates mobility requirements, policy and mapping rules by taking into account mobility prediction information. As the predictability of vehicular mobility is direction-oriented rather random, and it is easier to know where the MU is heading if the movement information is known in advanced. The available information includes the collected data during ASI subscription and MU movement history distributed over S+ access node and an information centric mobility prediction technique is being implemented. By associating MU current trajectory, speed, movement information with existing position predict the adjacent position and accordingly execute the identification of the available slices. An accurate prediction of user mobility provides efficient resource and handover management. This further enables the mobility scheme selection provided

by the slice configuration based on predefined rules. For dynamic re-mapping between mobility management schemes and current mobility requirements, the access control and mobility also perform the verification of the actuality of the mapping between runtime mobility demands for the given slice and adopted mobility management schemes and if needed performs modifications accordingly. ASI resource control functions provide the provision of network resources to virtualized network elements and apply negotiation between heterogeneous RATs for dynamic and customized configuration of resources. For example, it can request a priority based resource allocation for a given interval of time. Finally, the access level QoS mapping function enforces the translated policies, service requirements and targets KPIs on the slice instance. As described earlier in section 3.3 the EPS bearer concept in 5G is evolved towards Data Radio Bearers (DRBs) at AN and QoS flows at CN. In this context, this functional grouping manages the NAS level mapping between a flow to a proper QoS marking and the mapping of the QoS flow with a QoS marking to the DRBs. The downlink QoS flow decision is made on ASI access level and the uplink QoS flow will be mapped to the same DRB that carries the DL flow. For flows that are not reflective and no RRC configured mapping is available, the flows will be carried over default DRB. The Access level QoS will further illustrated in the later section 5.7.4 of QoS provisioning in automotive slicing that presents the detailed expression of proposed slicing scheme.

At CN level there are two control plane functional sub groupings i) ASI Session control and ii) ASI Policy control. ASI Session control referred to as vSF using 3GPP terminology. This functional sub grouping provides the configuration of the data plane to provide the traffic handling functionalities. The vSF functionalities also include selection and control of data plane anchor entity, allocation of IP address, data plane management in a distributed manner, support data forwarding for session continuity, re-selection of data plane function and data plane path, etc. The ASI Policy Control (vPF) provides policies for QoS enforcement, charging rules and signal QoS attributed between the network nodes.

The ASI data plane includes the corresponding virtual data plane functions (vDPFs) implemented as virtual switches at access and CN level to apply traffic processing characteristics. One or multiple distributed vDPFs can be activated and configured as needed for a given scenario. Based on the

actual 3GPP specification [12], for the processing of downlink traffic vDPF maps the traffic to QoS flows according to the Packet Detection Rules (PDRs), FPI, Max Flow Bitrate and Session Bitrate provided by vSF and implements the Session-AMBR received from the S+ DS. vDPF includes the QFI in the encapsulation header. The vDPF further implements the transport level packet marking on downlink traffic per QoS Flow basis using the value provided by vSF. The access level vDPF maps QoS Flows to access resources based on QoS requirements and the associated QoS profile. Similarly, for the processing of uplink traffic, vDPF uses the stored QoS rules to determine mapping between uplink traffic and QoS Flows. When passing an uplink packet from access level to core level, the access level vDPF performs transport packet marking on a per QoS Flow basis.

## **5.5 Automotive Network Slices**

The variety of automotive use cases categories affect the design and placement of ASI access and CN functions. To fully appreciate particular issues described in section 5.3 and based on the specific functional requirements as described earlier in section 4.3, I advocate the following set of automotive network slices as shown in Figure 5.4.

The slice instance for vehicular internet and infotainment applications is implied to use multiple RATs to achieve higher V2I connectivity capacity and higher data throughputs. Cloud computing techniques to host contents on a remote application server and/or mobile edge server is used to reduce the latency. The higher vehicle speed may either overload mobility control functionality for this slice instance and usage of multiple Access and Mobility Functions (MFs) may be configured. The slice for vehicle remote diagnostics system is configured to send and receive connected vehicles system information periodically to the remote server. For this purpose, data exchange functionalities should handle multiple interactions through multiple gateway anchor instances and session control functionalities should be instantiated accordingly. The slice instance for autonomous driving and safety critical services is configured for low-latency V2V and V2P periodic or event driven messages. This also relies on additional R(AN) functions, such as for network controlled resource allocation at access nodes, authentication, authorization, subscription and mobility management. Moreover, a V2X application server co-located at the access node to help

vehicles in 3D-map processing of the surrounding area and building an augmented vision beyond their visual perception.

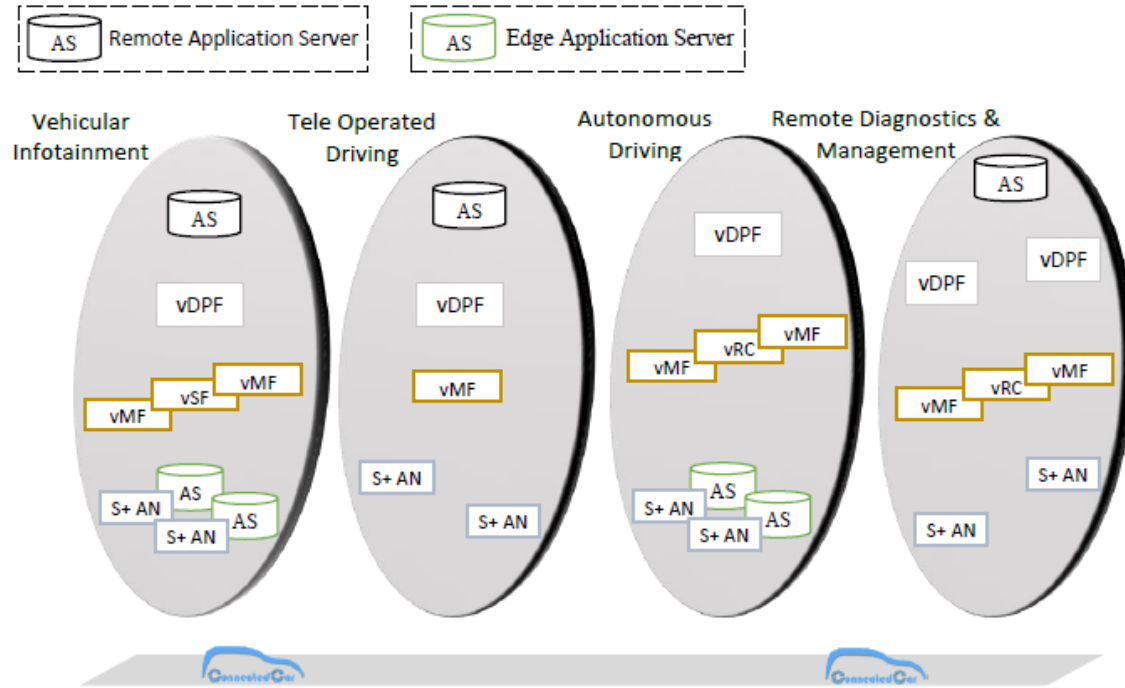


Figure 5.4: Overview of the proposed slice instances for the identified V2X service categories

The slice instance supporting tele-operated driving are instantiated only in particular circumstances and will be limited to a few vehicles. Such services result in light load on the network but must ensure ultra-low latency and highly reliable end-to-end connectivity between the tele-operated vehicle and the remote control room which is typically hosted outside the core network. The usage of multiple RATs may be configured to provide redundant connectivity to the tele-operated driving slice.

## 5.6 Exemplary Procedures

In the following, I discuss some exemplary procedures to further explain the proposed slicing architecture and elaborate how it can provide the flexible mechanism for slice instance selection, configuration and orchestration. It also elaborates the gain benefits of S+ slicing architecture such as significant decrease in signaling messages processed by control entities relative to current mobile network architecture.

### 5.6.1 Initial Attachment and V2X AS Connection Setup

Figure 5.5 shows the signalling diagram of initial attachment and V2X AS connection setup, the S+ AN receives an attachment request message (1) from MU which includes its capabilities, subscribed slice type and intended services. The S+ AN forward this request to ASO, where allowed and potential slice types are determined by interacting with S+ DS. The SCA selects the appropriate slice instance, namely the ASI-I based on the information received from the MU in the Attach Request and profile information from the subscriber data server. The slice orchestration forwards the AN registration message (2) to access control function, where the access control function (vMF) initiates MU slice specific authentication

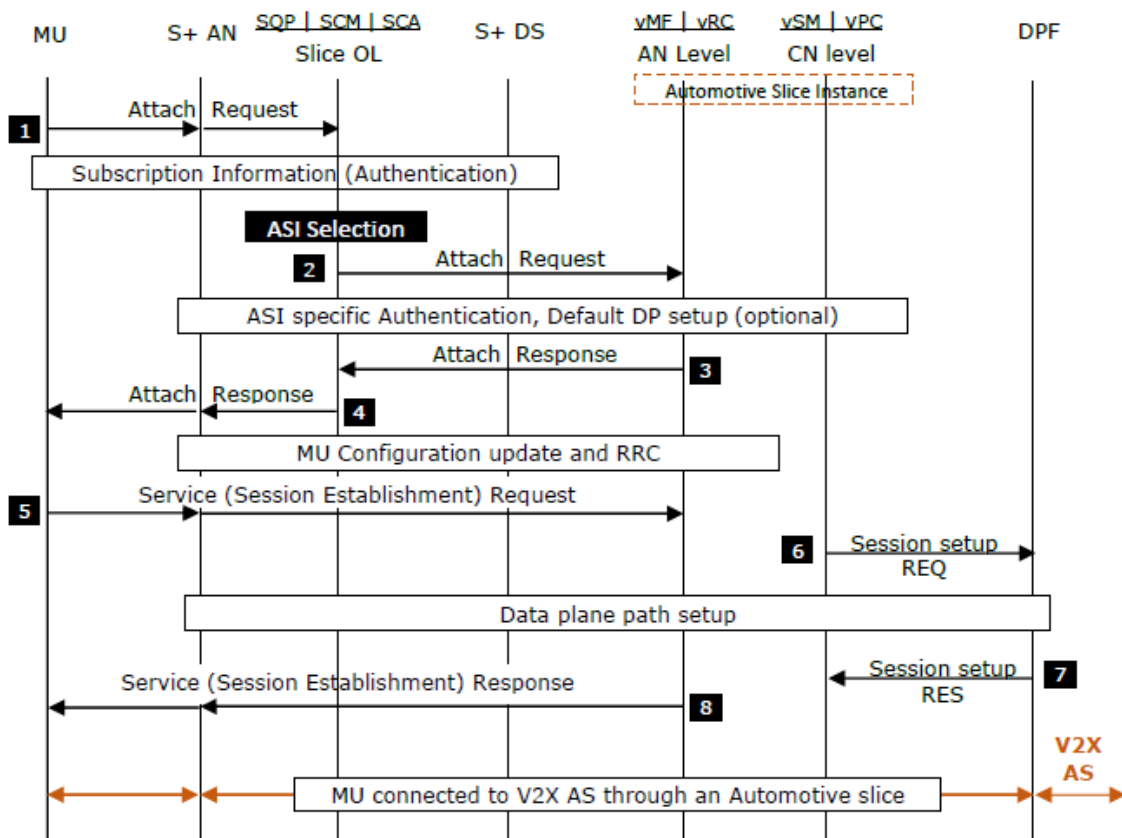


Figure 5.5: Signalling diagram of initial attachment and service setup

and carry from access network both access specific and access common information and interact with the corresponding control entity to provide the initial context setup. This may include optional setup of default data plane connection and the state of MU transitions from deregistered to registered state. After the successful slice specific authentication and authorization, an attach response (3) is send back to S+ AN which includes the slice instance ID and the MU Temporary ID. The S+ AN reconfigures the

radio connection, and forward the attach accept message to the MU. The S+ AN trigger the radio access bearer (RAB) setup procedure (4) and if no data plane connection is set up or if only a default data plane connection is set up for the MU and the MU sends service (session establishment) request (5) to the AN based on the MU temporary ID, which was allocated during the initial context setup. The AN then forwards the session setup request to corresponding vMF which interact with vSF (6) which performs the session management procedure to set up the data plane connection (7). In the case where the session requires a specific QoS policy, the vSF needs to interact with the vPF server through the service interface to get the corresponding policies before allocating the DPE. Furthermore, considering the expected handover with MU mobility, the vMF in cooperation with the vSF control functions notify the needed support for DPE relocation and keep the established connection context. At the end of this procedure, ASI sends the session setup response (8) to MU and a data plane connection is established from the MU to the correspondent V2X AS.

#### **5.6.2 Multiple connections setup to multiple Network Slices**

After the initial attachment and connection setup, the network's local conditions or policies may change, or an updated slice assistance information may be sent by the MU or the MU subscription information may be updated. These changes lead to a modification of the active set of slice instance, as the current configured vANFs or vCNFs may not be any longer fully suitable for the connected MU. For example, one set of slice specifically configured functions within ASI-I provides an enhanced mobile broadband service to the MU, whereas another set of functions within ASI-II provides a critical communication service to the MU. Implied such scenarios, Figure 5.6 models the service connection setup via multiple slices instances that allows steering of traffic into two isolated automotive slice instances comprising of specific data plane as well as control plane function while sharing common set of slice orchestration functions.

Based on the network policies or updated subscription information, the slice orchestration determines a new slice instance that is allowed for the MU. All control plane functions (CPFs) that are common to both slice instances, are not necessary to be created multiple times. If the common CPFs needs to be changed, ASO functions are responsible for the configuration and allocation of target common CPF. Other CPFs that are not in common with

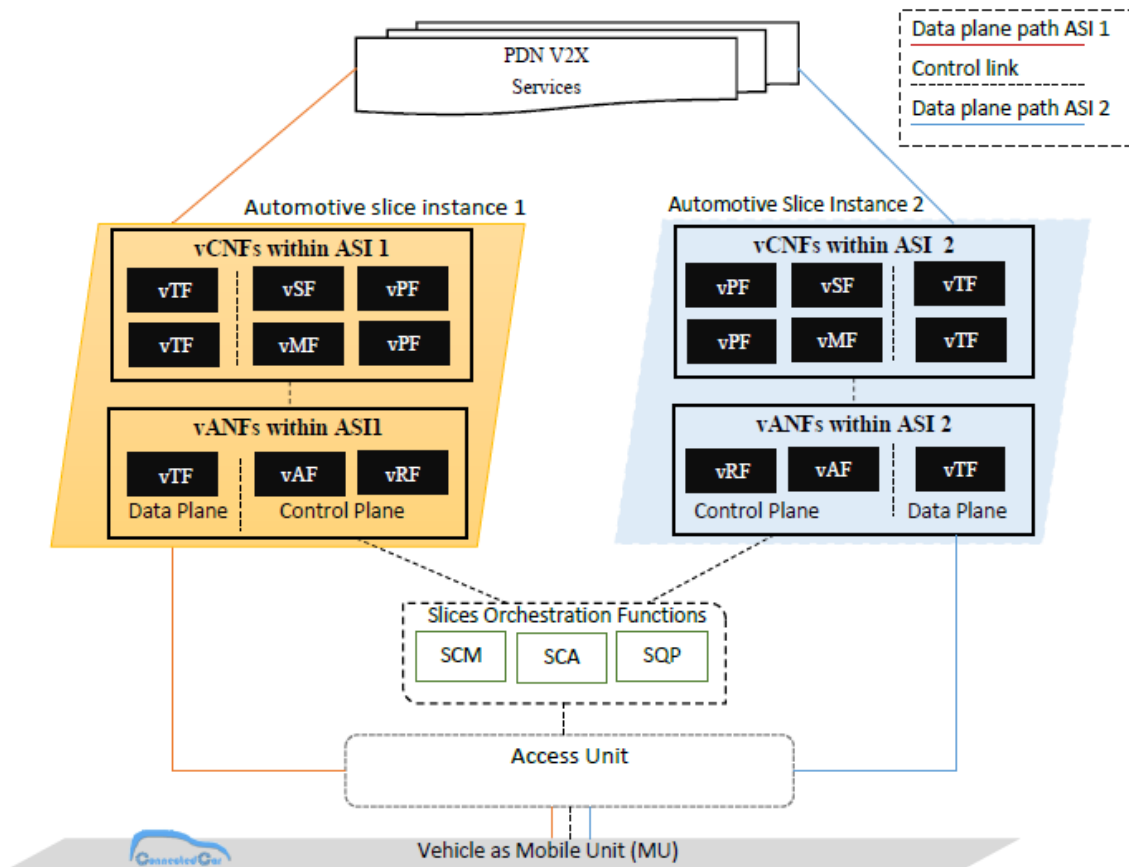


Figure 5.6: Service connection setup via multiple slices

other slice instances are only used by its own slice instance. The AN communicates with the ASO function or the CPF specific for ASI via a single control link interface regardless of which CPF within the ASO that the RAN is communicating with. A set of vDPFs in respective ASI is responsible for providing a specific service to the MU and for transporting the user plane data of the corresponding V2X service. The MU can have multiple data plane connections to different sets of vDPFs that are available at different slice instances simultaneously.

In the case of multiple connections set up through multiple slice instances, the new ASI selection is triggered by ASO with the following procedure. As depicted in Figure 5.7 the MU performs the V2X AS connection establishment over initial slice instance as described above in section 5.7.1, with the following additions. The MU includes additional information, e.g., service Type and or corresponding V2X AS name along with attach/service connection setup request (1), this information is used for identifying the V2X AS that a MU wants to communicate with and the ASI is supposed to support. There can be both standardized service type applicable to all automotive slices or slice specific service type. This service type will be

stored at the MU and can be updated according to the network conditions. The AN use given identity information to determine the appropriate vMF function and route the NAS message to corresponding vMF (2), where authentication and admitting the MU to connect the automotive slice instances 1 and 2 is performed and attach response is send back to ASO (3).

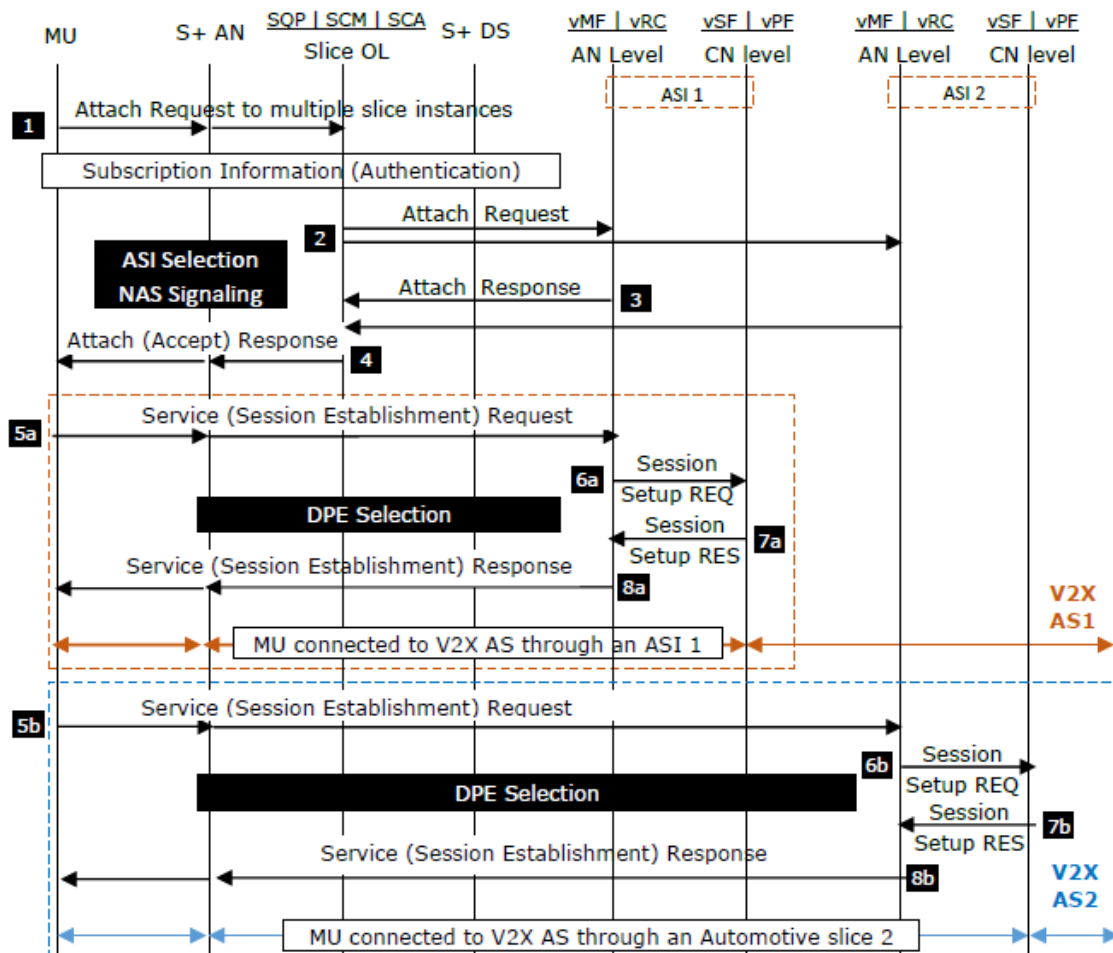


Figure 5.7: Signalling diagram for service connection setup via multiple slices

The SCA function sends multiple connections accept response (4) to the MU with the specific ASI-I CNFs and ASI-II CNFs, for which the MU is to be configured to attach. In this response, it contains the identity and the allowed ASIs information, for which the MU is to be configured, its corresponding service Type and/or corresponding V2X service-ID that the MU is allowed to connect. In case, the V2X service-ID newly provided does not match to the ones that the MU already has, the V2X service-ID(s) will be configured at the MU. After step 4 has been performed, the MU may start for service connection setup by sending service setup (session



establishment) request (5a) to the access unit. The MU includes ASI identity in this message which was allocated during the initial context setup. The access unit forwards the session setup request to ASI-I vMF which interact with corresponding vSF which proceed with the setup of data plane connection. Based on the transferred session identity the vSF selects a vDPF and sends the session setup request (6a) to the vDPF. After a successful session setup, the vSF sends the session setup response (7a) back to the vMF of the ASI-I. The service session setup response message (8a) is then send back to the MU via the access unit. After this step, the MU is successfully connected to ASI-I with an active PDN session.

In this case, the MU requests for the establishment of another session for a new V2X AS that is of a different service type than the initial communication service. In this service (session establishment) request (5b) to the access unit, the MU provides the MU identity information, usage type and V2X service type and/or V2X service-ID. The access unit forwards the session setup request to ASI-II vMF which interact with corresponding vSF which proceed with the setup of data plane connection. The vSF selects a vDPF and sends the session setup request (6b) to the vDPF by using the V2X service information in the session request. After a successful session establishment, the vSF sends the session setup response (7b) back to the vMF of the ASI-II. The service session setup response message (8b) is then send back to the MU via the access unit. After this step, the MU is successfully connected to ASI-II with an active PDN session.

### 5.6.3 QoS provisioning procedures

Figure 5.8 shows the control plane signalling, data plane aspects and the usage of context and policies for QoS provisioning in automotive slicing. A MU is connected to V2X AS through an automotive slice and an established data session carries all traffic regardless of the QoS characteristics of individual traffic flows. Given the different set of QoS characteristics by the diversity of the V2X services and traffic in conjunction with their increasing expectations, the task of QoS provisioning is approached from the perspective of information centric with the input from service requirements. As part of the connection setup, the MU is provided with a set of pre-authorised flow priority indicators using access-stratum signalling and it can initiate uplink packets without explicit signalling with the network.

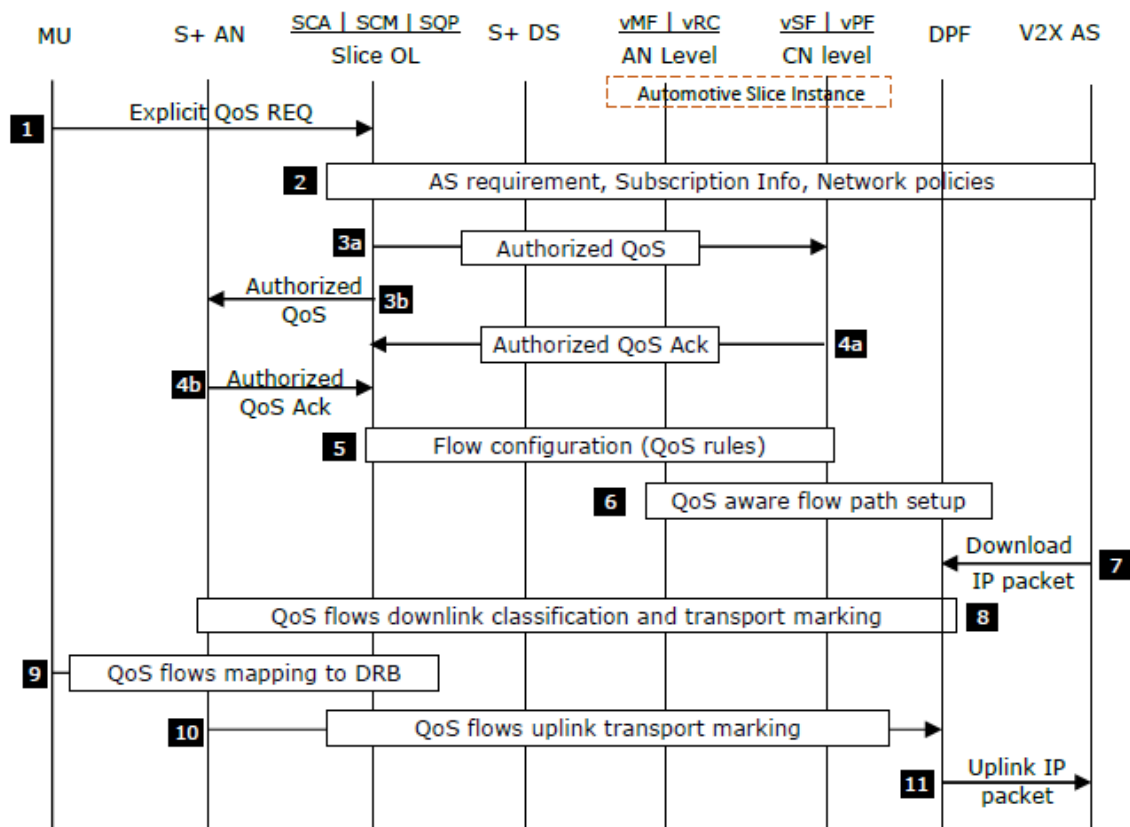


Figure 5.8: QoS provisioning signaling

For customised service requirements, an explicit QoS request is received (1) by the ASO functions from the MU. As described earlier in section 5.3 the SQP is the main application function within the orchestration layer responsible for QoS provisioning. The SQM first determines (2) the authorised QoS (flow priority, max flow bitrate, session bitrate etc.) with the information provided by the subscription terms, slicing policies and application QoS requirements received from MU. The real-time provision of such variety of information not only able to accurately calculate the additional physical/virtual resources required to maintain the automotive slice instance but also enable to derive the individual QoS aware configuration and take appropriate action to ensure service quality bounds.

By utilizing SDN, the SQP application function connect the slice instance virtual functions distributed in the AN and CN, mapping of the authorised QoS (3a). The SQP function also sends the authorised QoS message (uplink and downlink flow descriptor) to the AN (3b). The flow description is used to validate the flow priority indicator received within explicit QoS request by the ASO functions. In the step (4) the AN acknowledges QoS enforcement operation to SQP by sending an authorised QoS Ack message

(4a) to ASO functions. Similarly, the DPFs acknowledge QoS enforcement operation to SQP by sending an Authorised QoS Ack message (4b) to ASO functions. As part of this step, the ASO functions also provide information that allows the AN node to identify the PDN Session and notify the established resources corresponding to the QoS request. After receiving the acknowledgement SQP functions requests the specifically instantiated control functions (vMF, vSF, vPF) to define and handle the QoS rules (5). Depending on the type of request the flow configuration request is handled in step (6), the vSF determines the (high-level) rules for traffic flow configuration with input from vPF. The explicitly signalled QoS rules to vDPFs contain,

- The decided QoS parameters (e.g. MBR for an SDF, GBR and MGBR for a GBR QoS Flow). A unique QoS rule identifier is assigned to associated QoS flows.
- A defined flow descriptor and priority indicator, enabling DPF data traffic classification, bandwidth enforcement and transport marking.
- The QoS aware routing path details based on the instantiated DPFs topology.
- The packet handling rules to be transferred to AN and MU for QoS enforcement.

Upon reception of a downlink IP packet (7), the DPF identify the packet with downlink flow descriptor received within QoS policy. Based on the received flow configuration, the DPFs apply packet classification and marking of data plane traffic (8). The packet marking is used in the encapsulation header, and transport marking determine the QoS aware routing path. If the DPF determines that reflective QoS should be used for corresponding data traffic, it includes the Reflective QoS Indication (RQI) in the packet marking. In addition, the DPFs apply maximum bitrate control for downlink packets at the flow and PDN Session level. Subsequently, as the AN receives the downlink packets, the QoS flow mapping to Data Radio Bearer (DRB) is triggered (9) by the AN. If RQI was included in the packet, the AN replicate it in the radio header of the downlink radio packet. In this case, the AN also includes flow priority indication information in the downlink radio packet that assist the MU in determining QoS for corresponding uplink traffic. The MU upon reception of downlink radio packet identify the QoS indication information (RQI and/or Flow priority) and handle the corresponding uplink flows accordingly. An application service in the MU provides an uplink packet

for transmission to the lower layers. Based on internal configuration the MU maps the uplink packet into the pre-authorised flow descriptor and proceeds with the transmission in the uplink. If the MU currently has no related DRB for given flow description, the MU initiates a new DRB and includes the selected flow priority in the radio packet header. Upon reception of the radio packet, the AN (10) also check if included flow descriptor is part of the authorised QoS information. After verification, the AN performs mapping of the received uplink (GFB, non-GFB) values in the encapsulation header and determine the uplink transport marking (10) accordingly. Upon reception of the packet, the DPF performs validation of the flow descriptor. If the DPF determines that the flow requires a different QoS than the one selected by the MU, the DPFs perform a QoS modification. DPF forwards the uplink IP packet towards the V2X data service (11).

#### **5.6.4 QoS aware slice management and orchestration**

The proposed slicing scheme provides the capability of flexible mechanism for slice instance selection, control and management. Through the design and placement of specific access and core network functions in the respective set of automotive slices as described in section 5.4, the individual and personalized QoS is pre-emptive for different automotive use cases categories. Four important categories for automotive slices are proposed as described in section 5.5. After activation of given slice instances, the challenge is to ensure the QoS compliance of the services that are being supported by these automotive slices. This compliance is being managed and enforced by the slice orchestration layer through the interface protocols enables QoS aware configuration, allocation, coordination across different slices. Due to the individually created commands and recommendations per automotive use case category, the automotive slicing scheme has the capability to take into account the individual V2X service parameters and KPIs, such as the location, the speed, the available interfaces, the available network resources etc. to incorporate them into the orchestration decision. The generated commands and recommendations define or limit the inter-slice handover considering the individual QoS requirements. The individually generated commands and recommendations per end-user device help to optimise the traffic load essentially at the access networks by considering requirements of different V2X services.

An example of the QoS aware slice orchestration is illustrated in Figure 5.9. The connected vehicle (MU) moves on with an initial connection setup and provides the default automotive access as assistance information. The default automotive access depends on the type of V2X AS such as vehicular infotainment and is used by slice orchestration for selecting an automotive slice instance. For instance, ASI-I is configured for automotive internet and vehicular infotainment referred to as usage class A. The ASI-II is configured for vehicular infotainment and remote diagnostics system referred to as usage class B respectively. A V2X application for automotive internet is launched by the back-seat passenger in connected vehicle and initiates a new data flow. By using the usage class selection policy, the MU associates this data flow with usage class A through ASI-I. The MU requests the establishment of a PDN Session for usage class A in the PDN session request message. After establishing the PDN Session the MU routes all data flows of the V2X service through ASI-I that is associated with given usage class A.

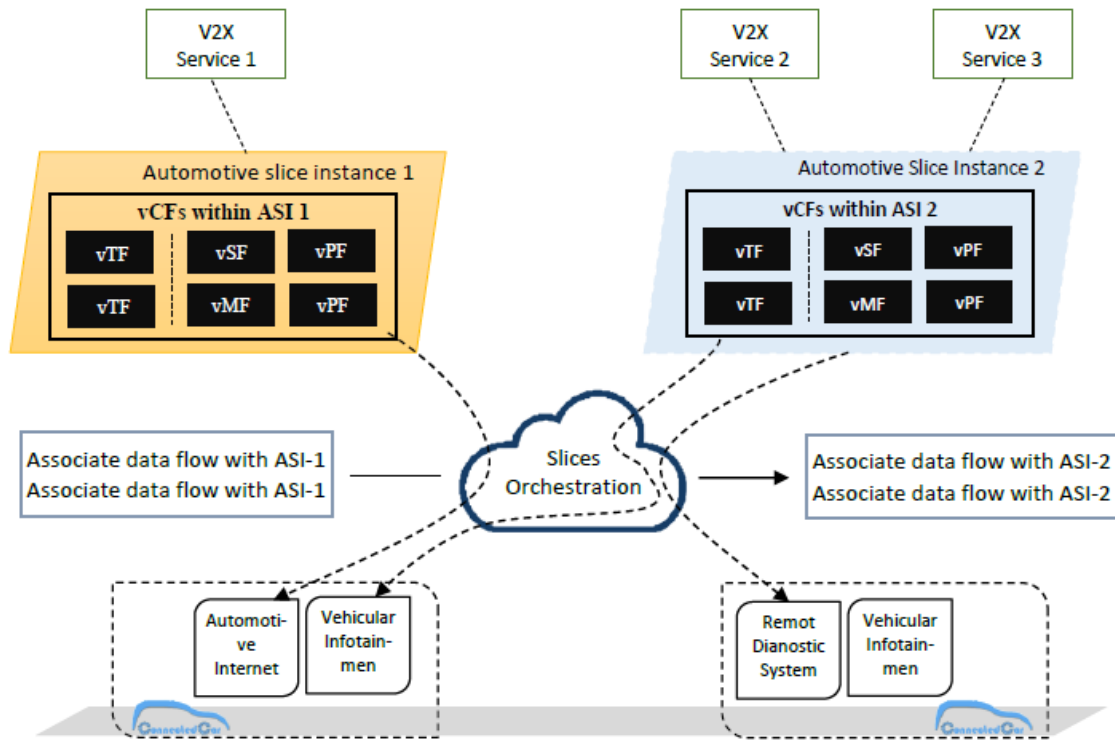


Figure 5.9: QoS aware slice management and orchestration

The Slice QoS provisioning (SQP) is the main functional element of slice orchestration layer that periodically monitor the instantiated vNFs of automotive slice instances and has the granularity to observe each individual resource usage assigned to individual vNFs making up the

automotive slice instance. As it detects that provided service is not in use or service degradation, the slice orchestration is provided with the performance report in recognition of the specific event. Later a remote diagnostics application is launched in the connected vehicle and initiates a new data flow that is associated with usage class B. Since the MU is not connected to a slice instance that supports usage class B, the MU provides to the network the usage class B as "assistance information" to be connected to another network slice instance. The ASO manage the connection of MU to new network slice instance that supports the usage class B. After the MU establishes a PDN session for usage class B and routes all data flows of the V2X service 3 applications associated with this usage class B within the PDN session. As a result, individual and personalized actions are driven to determine the most suitable slice instance. This example just serves as an illustration of how enhanced ASO layer has orchestration decision depending on the target usage class and the requirements that have to be fulfilled. An outline of QoS aware routing during connection mobility, a generally more complex orchestration logic is illustrated in the next sections of IP mobility management.

#### **5.6.5 Mobility Management procedures in automotive slicing**

Many important issues related to mobility management has been extensively studied in the legacy network of entities and using SDN models being tested in experimental setups. However, mobility control and management in the slicing-based network of functions has not yet been actualized. Within a slicing-base network, the capability of flexibly supporting mobility of users and their connected MUs, as well as sessions and flows and even (virtual and physical) network entities is predominantly seen as a challenge in terms of changing performance and connected vehicles perceived QoS. As described earlier I proposed specific configurations of network functions and instantiate them according to given requirements. Similarly, for mobility management, I also maintain specific, mobility related flavours of network functions and/or specific configurations of network functions and instantiate them according to the mobility related context of an automotive slice. The selection of appropriate mobility management scheme provided through slice orchestration and dedicated functionality to support service tailored mobility control and management. In next I discuss a few mobility scenarios used to explain the behaviour in the inter-RAN handover, inter-slice handover with and without data plane

function re-location and describe the important call control flows being exchanged across the various network elements of the proposed solution to show how it can be systematically exploited to gain benefits.

#### **5.6.6 Inter RAN Handover with DPF relocation**

As shown in Figure 5.10, the MU is initially connected to a V2X service and data plane path is established through an access unit (hereafter referred to as source access unit) and DPF-1. The general system information is sent frequently to the connected MU by the AN over the specific system information broadcast channel. The MU also performs measurements on several parameters, depending on the AN connected to, and sends measurement reports to the network. In a pre-handover situation, the proposed (functional grouping) cooperative context acquisition, where the MU fetches network cognition from access Mobility anchor Function (MF) and initiates terminating cognition. Given the use case high mobility and able to fully exploit spatiotemporal received signal strength (RSS) diversity when moving all around the AN that leads to an improved detection performance. In addition, the AN and MF cooperation enable dynamic radio information and policies and real time available access networks information with policies, in accordance to priority scheme are provided towards the MU. I followed the network controlled terminal assisted decision mode and the serving MF perform the inter-RAN handover decision based on the contexts and policies. In addition, based on the distribution of DPFs topology adaptation and (Re)selection of efficient data plane paths, the handover decision may also contain DPF relocation instead of simple RAN handover. The handover decision is sent as individualized recommendations to the connected MU. The MF identify the presence of an appropriate local access unit and it also determines if the initial access unit provides interworking services with the target access network and send a handover command (1) to source access unit. The source access unit triggers the forwarding of data message (2) to the target access and the path switch request (3) is sent to the targeted access unit. The MF, in turn, forwards this session establishment request received from the connected MU to slice instance vSF. In given case the vSF need to select a new DPF with following specific considerations a) Capable of combining multiple simultaneous connections of MU via multiple access nodes. b) The MU may need to access a single PDN service or access both a new and central PDN service. c) MU releases the existing PDN session associated with an initial DPF and immediately

establishes a new PDN Session with a new DPF to the same or new PDN service. d) The multi-homed PDN session to be used to support seamless session as well as service.

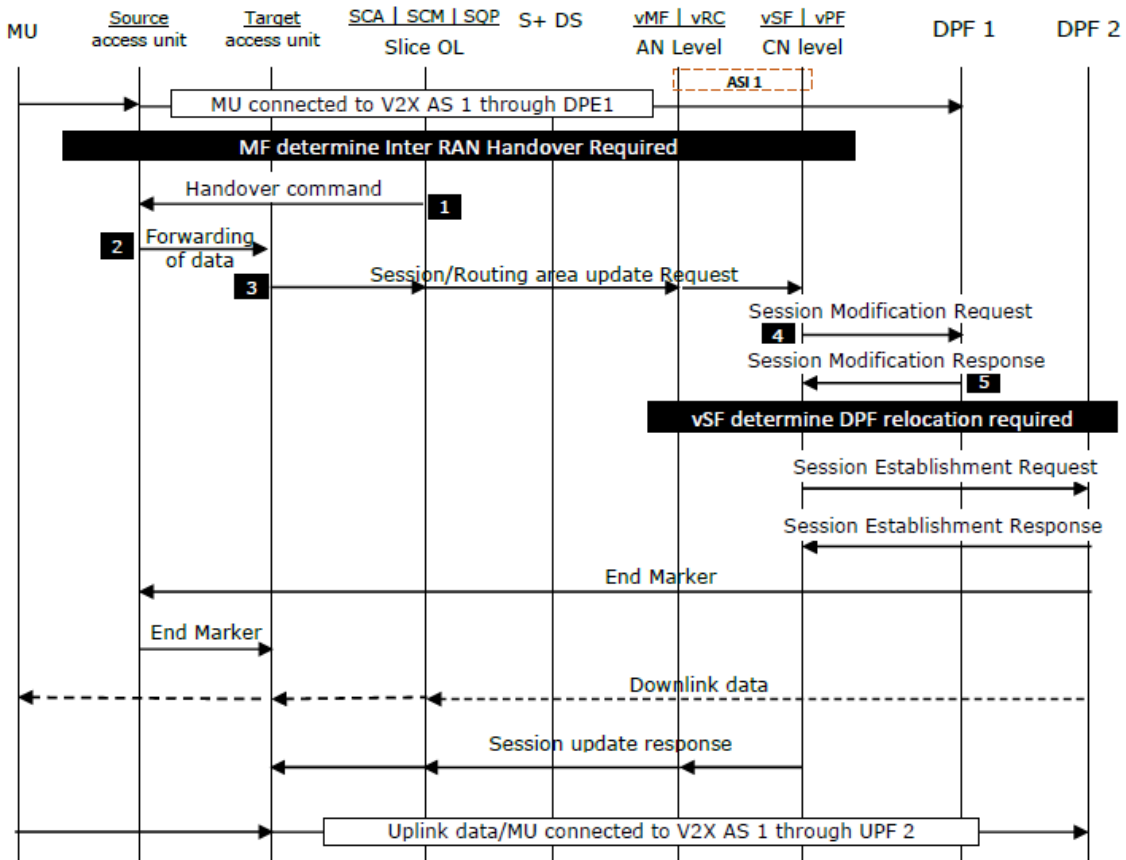


Figure 5.10: Inter RAN Handover with DPF relocation

In case such as in an intersystem handover where simultaneous access to a single V2X AS via heterogeneous access networks is needed and vSF decides to allocate the current session with a new DPF operating here also as IP anchor. In the process of selection of new DPF for the multi homed PDN session a new IP prefix is allocated for the PDN session. This implies a key role for IP mobility management in providing support for different IP flow modes.

#### 5.6.6.1 IP Mobility Management

There have been several approaches for IP mobility management in SDN based mobile networks, as surveyed in [98], [99], In, [98] the authors applied the SDN concept to DMM architecture for routing optimization with a DMM service within SDN controller. In the proposed solution, when a user attaches initial router, the SDN controller stored user information in Binding



Cache Entry (BCE). If the user moves to the new router, the controller which receives the packet-in message will check BCE. Mobility management is supported to the user by binding update with DMM service. The new data path is set by SDN controller and it sends Flow Modify message to previous and new router. On receiving flow modify messages, the routers will update their flow tables. That is, the user can be supported optimized path by flow table, without tunnelling. But It is noted that after handover, first, the packets have to follow the route from the previous to the new router. The packets are then redirected between the new router and finally to the user. Consequently, it may lead to a complex process and a high signalling cost. In [99], the authors considered two schemes for mobility management in DMM scenario called Tunnelling mode and Optimal mode DMM, in first scheme different prefixes are allocated to the user at old and each new attachment nodes. For the ongoing flows which require mobility support, the list of old and new prefixes are included in packet-out message named router advertisement message. The controller updates the forwarding rules at both old and new OpenFlow switches (OFSs). Using the tunnelling mechanism the flows are being redirected between distributed OFSs. In the second scheme, the controller calculates the new route for the on-going flows and populates the new forwarding rules to all the intermediate OFSs along the new route between the new OFS and the correspondent node. In other words, an explicit path between the user node and CN is defined and established by the controller. In this way, a tunnelling mechanism can be avoided. However, it may lead to a complex task giving the large number of switches/flows to handle. Besides, the simplicity of the first scheme can come at the cost of tunnelling overhead and sub-optimal routing. The authors indicated that the optimal mode is likely more suitable for latency sensitive services while the tunnel mode seems to be better for the packet loss-and interruption-sensitive services.

In line with these directions, to fully appreciate the particular challenges of QoS aware routing associated with IP mobility, I further elaborate the connection mobility in providing support for different IP flow modes in the following cases. i) The MU requires a static IP address for incoming IP flows. Static anchoring at the home DPF will be required. ii) No fixed IP address, i.e. the MU acts as client, but IP session continuity: No static anchoring at the home DPF will be required. iii) No fixed IP address and no IP session continuity: no static anchoring at the home DPF will be required. Moreover,

the solution implies additional support from the different network entities and functions. Such as MU need to indicate support and request for IP flow mobility mode to the ASI when access is added for a PDN connection. Similarly, the MF also needs to be capable to provide an indication to the vSF for its support for IP flow mobility during the PDN connection establishment over the first access and over additional access. When the vSF receives IP flow mobility indication, the vSF needs to be capable to negotiate the support of requested IP flow mobility and confirm the IP flow mode. The vSF also support a PDN connection using multiple accesses and exchange routing rules with the MU over control plane protocols as well as receive notification that access has become "usable" or "unusable". In the following, I discuss the use of IPv6 multi-homing, and MPTCP separately or using IPv6 multi-homing together with MPTCP to support the simultaneous exchange of IP traffic to a single or multiple V2X services.

#### **5.6.6.2 IPv6 multi-homing only seamless connection mobility**

IPv6 multi-homing represented connection mobility procedure is illustrated in Figure 5.11. Using the 3GPP terminology the DPF of the given scenario hereafter will be referred to as PDU Session Anchor (PSA). When the vSF within a serving ASI determine that the PSA needs to be changed, the session requirements are invoked, and session modifications are prepared accordingly. The vSF selects (2) a new DPF and configures it as a new PSA for the multi-homed PDN session. A new IPv6 prefix is allocated in this process and the PSA relocation is performed as described in [22] with following differences. The solution implies AN level DPF being establish as Branching Point (BP) or Uplink Classifier (CL) by vSF (3). This provides forwarding of uplink traffic towards the different PSAs and merges of downlink traffic to the MU i.e. merging the traffic from the different PSAs on the link towards the MU. Furthermore, the implementation of transport marking at AN level is done as in BP or Uplink CL that enables optimal data path relative to the initial and new PSA. Next, the MU is notified of the availability of the new IP prefix using an IPv6 router advertisement message. The vSF updates PSA (4) according to application needs and sends routing rule along with the IPv6 prefix to the MU also using an IPv6 router advertisement message as described in [100].

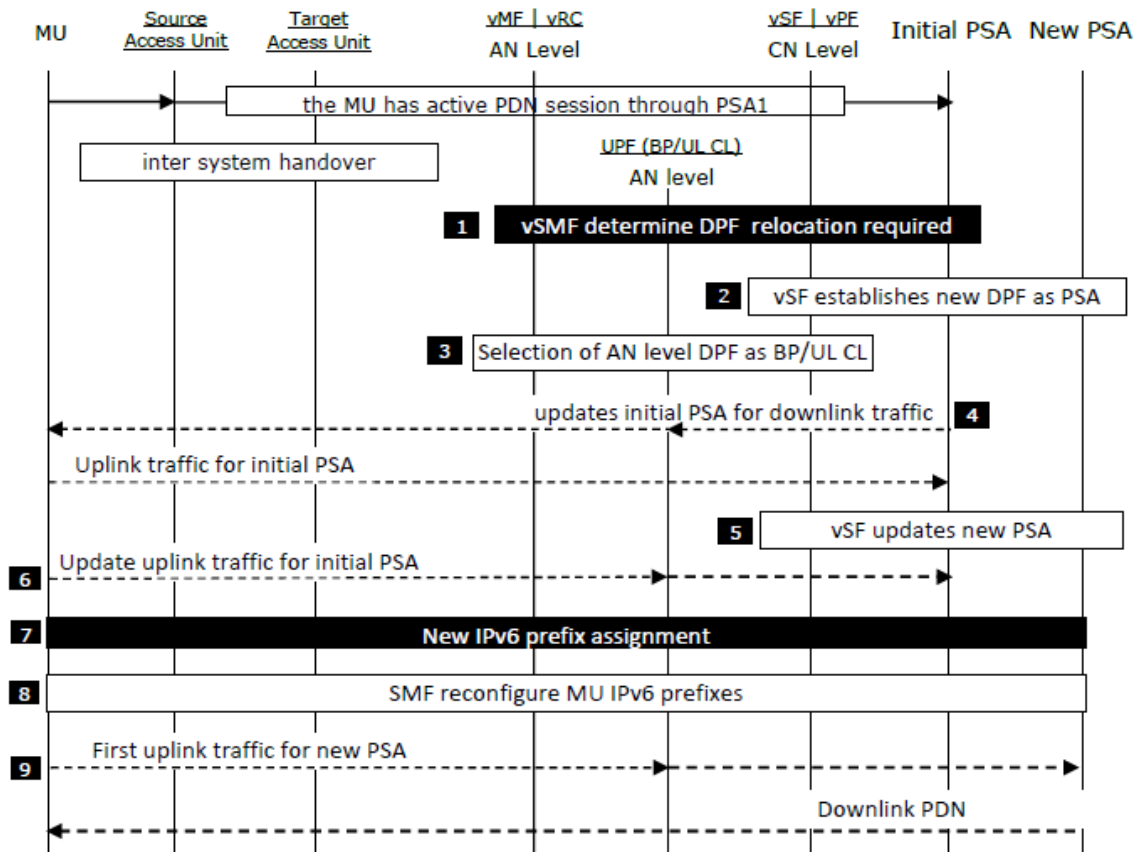


Figure 5.11: IPv6 multi-homing only seamless connection mobility

It also provides the BP or uplink CL DPF with the necessary uplink traffic forwarding rules related to the prefix of the IPv6 source address of UL traffic. The MU starts using the IP prefix associated with the new PDN session for all new traffic and also proactively move existing traffic flow from the old PSA to the new PSA. After the new session setup process is completed and as no traffic is carried in the initial PSA during the timer interval. To release the BP, the vSF updates the PSA-2 providing the DPF CN tunnel info and also updates the exist existing DPF instead of updating the (R)AN and releases the corresponding IPv6 prefix.

### 5.6.6.3 MPTCP only seamless connection mobility

The operation of the MPTCP-based connection mobility is shown in Figure 5.12. In this case, when the connection starts, MPTCP options are included in the SYN segments to assure that the user and corresponding PDN Service are MPTCP capable and a unique identifier called Token is also included to be used to link the additional connections. Once an MPTCP connection is established, each endpoint knows one of the IP addresses of its peer. During the movement of MU several events occur (e.g., handover from one AN to

other, selection of relevant MF and so on) that tag along with data plane IP anchor relocation and it must connect appropriate DPE (referred to as PSA in Ipv6 based multihoming solution) that is topologically close to the MU as an additional MPTCP subflow. Together with this procedure MU gets a new IP address and send a SYN packet with a JOIN option to the PDN service known address. Upon reception of this option, the MU will establish a new TCP subflow to the newly received address.

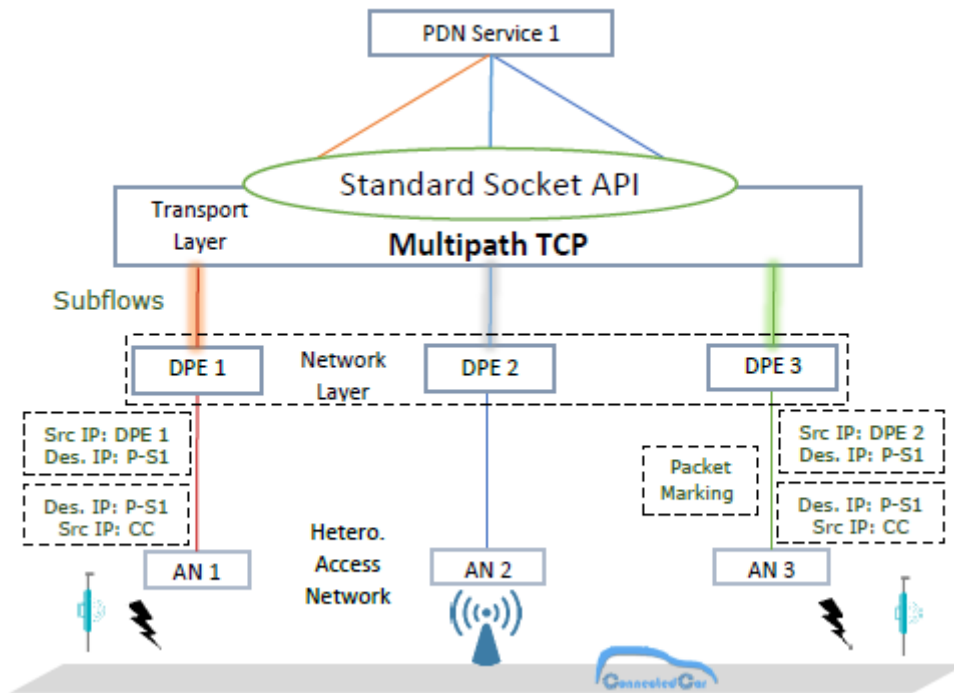


Figure 5.12: MPTCP based operation for connection mobility

To ensure the continuity of active communication and to prevent MU from idle mode, MPTCP support “make before break” method and MU initiates new subflow, in the same way, the MP\_CAPABLE handshake and sends SYN segment through new DPE with its new IP address to PDN-Service1. The new subflow needs to differ at least one of elements of four-tuples (MU IP address, PDN-Service IP address, MU port and PDN-Service port). With pre-included a local identifier (token) carried as an MP-JOIN option of SYN segment, both MU and PDN-Service1 are linked to existing MPTCP connection. After the subflows has been established the PDN-Service1 will be able to synchronize the user’s data traffic using different IP addresses distributed on MPTCP subflows. Recent enhancements to Multipath TCP path manager delegates the management of the paths to the applications [101]. This path manager enables applications to control how the different paths are used to transfer data. Furthermore, defined by MPPRIO option MPTCP

support different flow modes, in the single-path mode, only one TCP subflow is used at any time or using all subflows simultaneously between two communication nodes or uses only a subset of subflows for transmission of data packets. With successful communication over new subflow, the subflow of long route/delay is set as a backup with MP-PRIO option. The old DPE checks the MU activity, as no traffic is carried in the initial subflow (during a time interval) it starts the releases procedure for the removal of the initial IP address from the PDN-Service IP list. The MU could generate a FIN segment/RST flag to close a subflow. Unlike regular TCP that does not allow to send an RST flag when the connection is in a synchronized state, in MPTCP the RST flag has only the scope of subflow and only close the related subflow but not affect the other subflows. The release cause tag along with the session management and perform the DPE binding information update.

#### **5.6.7 Inter slice Handover with DPF relocation**

There can be several scenarios for slice change mainly due to network maintenance issue, MU mobility, subscription or policy change and/or service change. I describe this inter slice handover scenario happened when a MU moves to a new service area and it must connect through new slice instance. Moreover, depending on the given scenario inter slice handover could be MU requested or network triggered. As shown in Figure 5.13, the MU is connected to a V2X service and data plane path is established through an ASI-I. Due to mobility the MU or the ASO determine the needs to change the network slice which the MU is currently connected. In given case inter slice handover includes both the access unit and the DPF relocation. The slice orchestration identifies the presence of an appropriate slice instance ASI-II hereafter also referred to as target slice instance. Both source and target slice instances are independent, and they do not need to share any slice specific AN and CN level functions. However, they share common S+DS as the subscription database. The handover decision is sent as individualized recommendations to source access unit that triggers the forwarding of the data message to the target access unit and access unit relocation is performed as described above in section 5.7.2. Following the access unit relocation, the source slice instance forwards the (Routing Area Update) request message to target slice instance.

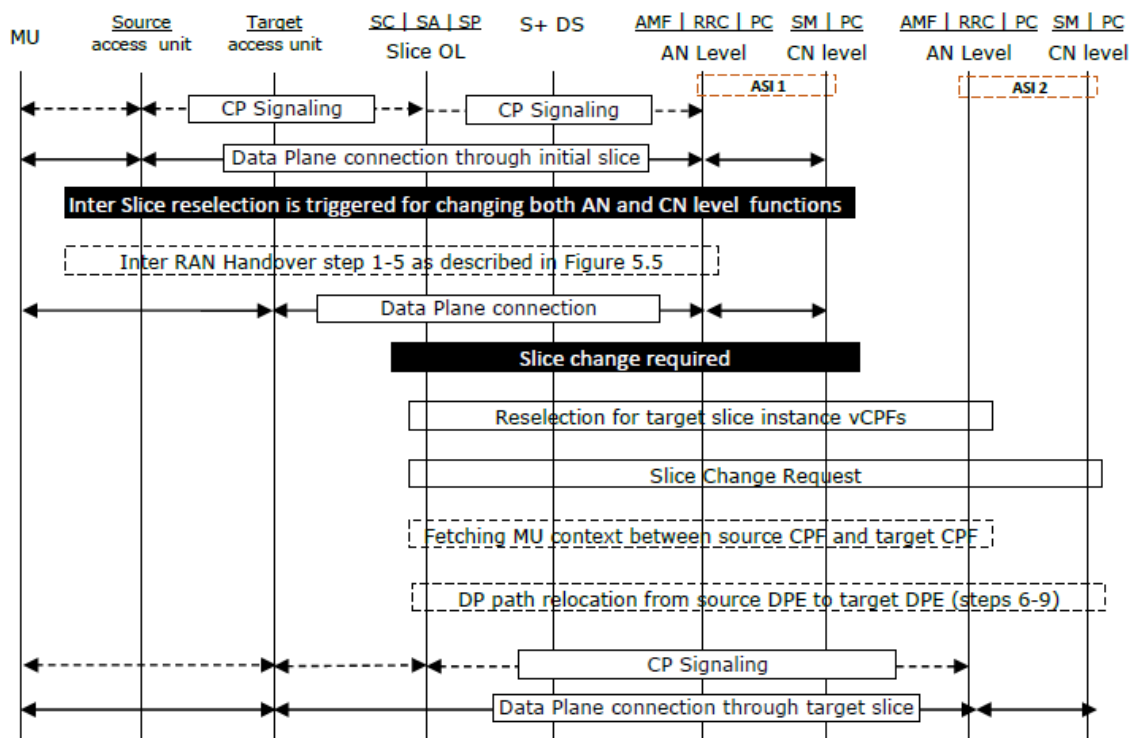


Figure 5.13: Inter Slice Handover

The target slice instance determines if the MU is allowed to receive services from the new routing area and sends a (Routing Area Update response) which indicates if the routing area update is accepted or rejected for required usage class. In case of confirmation, the ASI-II access unit forwards the session setup request to ASI-II MF which interact with corresponding vSF which proceed with the setup of data plane connection. In this request the MU identity information, usage type and V2X service type and/or V2X service-ID are provided. The vSF selects a vDPF and sends the session setup request to the vDPF by using the V2X service information in the session request. After a successful session establishment, the vSF sends the session setup response back to the vMF of the ASI-II. This response message is then send back to the MU via the access unit. After this step, the MU is successfully connected to ASI-II with an active PDN session. After the inter slice handover process is completed the target access unit sends a handover complete message to the source ASI-I, where the access resources are released. This notification about the end of the handover process enables the resource release right after the handover is completed and as a result, a waste of resources can be avoided. This contributes to releasing the relevant radio resources within the required time period and limiting the QoS degradation of the served connections.

## 6 Evaluation

This section describes the simulation environment we set up with the aim to assay the following questions:

- Assay the logical and structural design of the proposed S+MN architecture. Is the designed S+ MN architecture methodically executable and implementable?
- How it can be systematically turned to practical account to gain the investigated and defined automotive use cases operational needs?
- Does the application of the proposed evolutions improve the procedures of current mobile networks such as service setup, inter-system handover and the data plane management in a distributed manner, etc?
- Verify, how the proposed evolutions simplify the current mobility management procedures, QoS provision and cognitive resource management?
- To show, how to instantiate of multiple Data Plane Entities (DPEs) and demonstrate the forwarding of IP traffic through the use of distributed DPEs?
- Experimentally prove, how feasible is the use of Multipath TCP for connection mobility and path management in inter-system handover scenario in the following cases? i) The MU acts as a server and requires a static IP address for incoming IP flows. Static anchoring at the initial access node will be required. ii) No fixed IP address, i.e. the MU acts as client, but IP session continuity: No static anchoring at the initial access node will be required. iii) No fixed IP address and no IP session continuity: no static anchoring at the Initial router will be required.

### 6.1 Simulation Phase 1: Functional Validation setup

Among the emulation tools, ns-3 [102] is the most prominent network simulator that has long been used by the research community to test and develop networking protocols and services. The ns-3 is an open source, discrete-event network simulator for Internet systems, licensed under the GNU GPLv2 license, and is publicly available. Being written in C++ programming language, it can be interfaced with external libraries and tools from open source development repositories. The ns-3 can be built and used

with a C++ compiler, a source code editor (such as Eclipse, emacs, vim) and Python scripts can be written to interact with the ns-3. Moreover, ns-3 supports multiple mechanisms to validate the implemented simulation modules such as create virtual hosts, modify network nodes and verify tests and traces.

An E-UTRAN module (eNodeB, UE) and an EPC module (MME, S-GW, PDN GW) were implemented in ns-3 in 2011. In the following years, ns-3 also hosts modules for Wi-Fi, WiMAX, virtual device visualizer, mobility, spectrum and were continuously developed further under different projects. From ns-3.22 additional support for modelling of vehicular Wi-Fi networks (WAVE) has been added, including the channel-access coordination feature of IEEE 1609.4 [103]. In addition, a WAVE-compliant physical layer (at 5.9 GHz) and related statistics-gathering classes have been added to the wave module. Recently, an implementation for the simulation of 5G mmWave for the ns-3 is presented in [104], A basic module for the simulation of mmWave devices, PHY layer, MAC layer and channel models are developed. The implementation was done by the Cornell University, NYU WIRELESS academic research centre incorporation with Simons Foundation, and is publicly available at [94]. 5G mmWave module for ns-3 is fully customizable and allow to use it flexibly and analyse various scenarios with different settings such as carrier frequency, bandwidth, frame structure, etc., describing the behaviour of the mmWave channels and appliances. The details of all ns-3 modules implementing different technologies are available on [106] ns-3 documentation pages. The 5G mmWave module provides support for a wide range of channel models along with the recently included 3GPP FR1 models for frequency bands above 6 GHz. The PHY and MAC classes are parameterized and highly customizable and are flexible for testing different designs. For architectural exploration, it also includes emulation of main core network nodes and protocols such as complete emulation of the mobility protocols supporting intrasystem as well as intersystem handover, dual connectivity etc. The implementation of mmWave full-stack end to end module [105] enabled to conduct the TCP protocol performance analysis over mmWave bands and to integrate some novel networking techniques for optimal use of the available frequency bands.



To enhance ns-3 with SDN technology support an OpenFlow module, also known as the OFSwitch13 module [107], was designed, enabling open switch device as well as controller application interface to the ns-3 simulator as illustrated in Figure 6.1. The ofsoftswitch13 library is provided for OpenFlow switch device implementation and it is enabled to connect with ns-3 models by existing Carrier Sense Multiple Access (CSMA) network appliances and channels. The basic functionalities for controller implementation are provided by an OpenFlow 1.3 controller application interface that can manage a collection of OpenFlow switches. The OpenFlow channel interface is used to interconnect the OpenFlow controller and each switch. The standard ns-3 channels and devices can also be used to create an OpenFlow channel by using a single shared channel or individual connections between the controller interface and each switch device. The controller configures and controls the switch devices through this interface by receiving events from the switch and sending commands out the switch. The ns-3 TapBridge module [108] also enabled an external OpenFlow controller to interact with the simulated environment over this control channel running on the local machine.

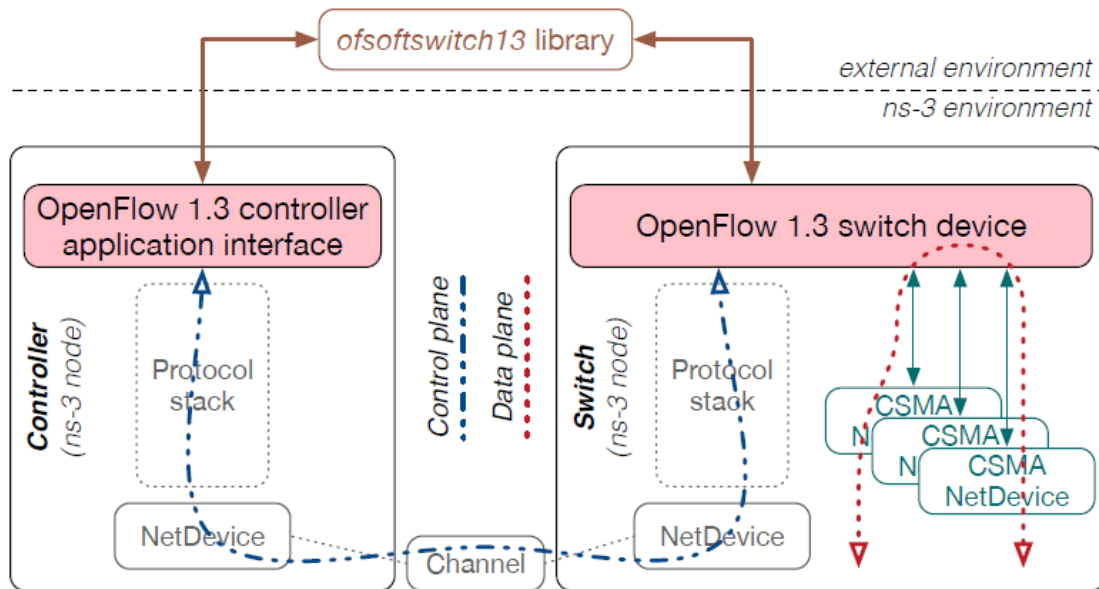


Figure 6.1: The OFSwitch13 module overview [25]

The ns-3 provides multiple methods of tests and traces for validation and quantitative analysis of the implemented simulation modules:

- Build verification tests are designed to assure the working of the software build.
- Unit tests that provide comprehensive assessment checks to establish the accuracy of the implemented source code.
- System tests are used to ensure the interworking of multiple simulation modules.
- Performance tests are included to test if a specific simulation part executes in an acceptable time frame.
- ns-3 tracing capabilities include very detailed framework allowing the user at different levels to personalize the extracted data from simulations.
- Examples simulation setups are provided that generate traces to verify the interaction between different simulation modules. These traces can also be used to validate if a simulation module is accurate and compatible with the actual model.
- Having provided, there are multiple levels of helper functions. The high-level helper function enables the user to manage the aggregation of pre-defined results to a fine granularity. The mid-level helper functions allow to personalize the data extraction and storage. The low-level core helper functions allow an expert user to update the actual model and to submit the updated and previous data to be accessible for other users at higher levels.
- Ns-3 logging method (NS\_LOG) allows to write extracted information into a log file. This information can be further analysed with different parsing scripts for example through the Linux grep, sed or awk commands to parse the information.
- Packet Capture (pcap) files can be produced in ns-3 to analyse the network traffic. Consequently, the generated pcap files allow to evaluate the data packets with network packet analyser software such as Wireshark. Similarly, ascii files can also be produced that consist of similar information as pcap files but in ASCII format.

## 6.2 Realization of S+ MN Architecture

The logical network topology of the S+ MN architecture realization scenarios are the ones shown in Figures 6.2 and 6.3. It represents the S+ access network nodes (S+ NodeB and S+ AP), OpenFlow switches entities (OpenFlow Access Switch and OpenFlow CN Switch devices), S+ NC functions, subscription and content server named S+ Data Server, client

and server sink applications meant to be used in tests. To realize the S+ MN architecture the main ns-3 modules are LTE, point to point helper, network, applications, mobility, core and OFSwitch13 modules. These modules have dependencies on other simulation modules and several modifications are needed in the existing version of the ns-3 to enforce the policies and the behaviour described in section 5.1. I extended the ns-3 environment to implement the following components, needed for the evaluation. Class diagram of OFSwitch13 module enhancement depicted in Figure 6.4. The source codes of implemented modules can be found in [109].

For the realization of the S+ AN, an access network comprising a 3GPP compliant S+ NodeB and a non-3GPP AP connecting to an S+ Core Network is implemented. The client applications which generate or consume data plane traffic also referred as Mobile Unit (MU) and triggers the S+ MN control procedures such as service request. The NAS signalling connection between the MU, the S+ ANs and the Access and Mobility Management Function (MF) to network access selection and control have been implemented within this research as well. This means for the MU to connect to S+ CN and includes access network identification, user authentication, authorisation, network selection, access barring and connection management etc.

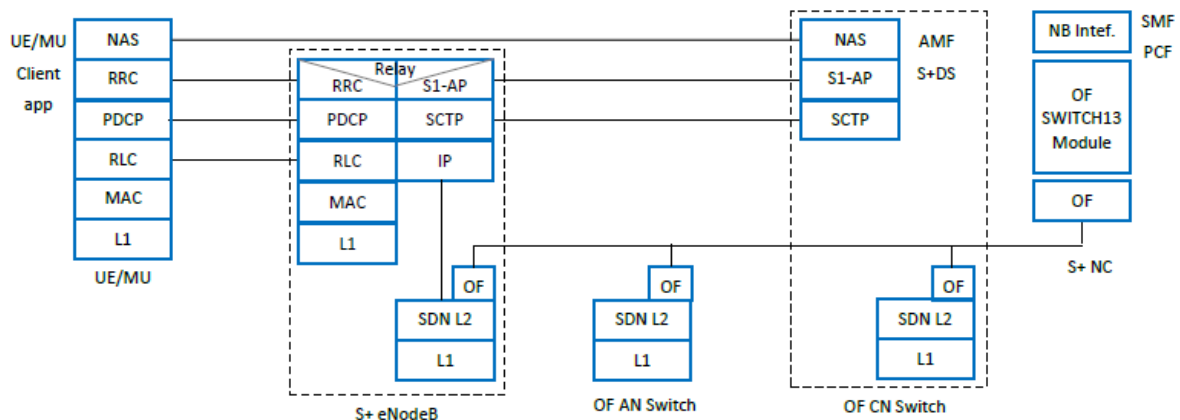


Figure 6.2: Control plane instantiation within S+ MN architecture.

The MF control function interacts with S+ Data Server to create a local copy for user authentication and authorization. I modify the code of access nodes (S+ NodeB and Wi-Fi AP) such that they can send the messages to OpenFlow Access switches which in turn relays the messages to S+ NC for data path configuration and session management. The evolved S+ access nodes are connected to OpenFlow Access and CN soft switches with existing CSMA channels, which acts as the intermediary between the S+ access nodes and S+ NC.

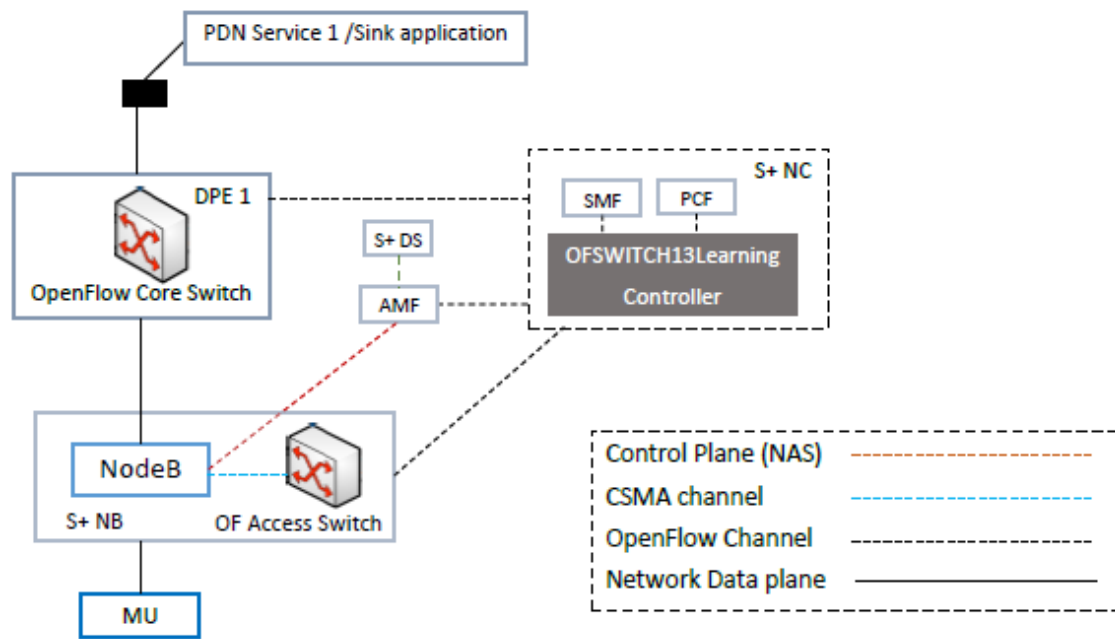


Figure 6.3 : The OFSwitch13 module overview [25]

To realize the S+ NC functions I use OFSWITCH13LearningController provided by OFSWITCH13 module in ns-3 simulation. To implement the control messaging procedures between the S+ NC and OpenFlow switches entities standard ns-3 supported OpenFlow channel interfaces are used. Through this interface, the S+ NC configures and manages the OpenFlow switch entities, receives events from the switch, and sends response commands out to corresponding switch device. To realize the mobility support in the considered scenario, some changes are necessary such as adding new processes and extension of the functionality of some previously developed procedures. Specifically, the operation and update of the CN switch device hereafter referred to as the Data Plane Entity (DPE) during inter-system handover procedure need to be implemented within the ns-3 environment. I imply coordination between MF and S+ access network to periodically collect and analyse reports about the network state. Assuming

that S+ NodeB serves only one cell, and a handover procedure needs to take place when a user performs a cell change while being in registered and connected states.

I extended the ns-3 environment to implement instantiation of multiple DPEs and modified an existing handover implementation in ns-3 to enable distributed DPEs support. It implements the messages exchange between the different network functions for intersystem handover and DPE relocation addressing the policies and the specific issues described in sections 5.1 and 5.7. The S+ NC calculated the forwarding rules for both initial and new DPEs. Additionally, I make S+NC to manage different types of IP flows and bind the new connections with their initial connection. In this way, the S+ NC has the full information of the initial connections and their new connections. Having provided, the IP address assignment helper of ns-3 enable the initial connection and the new connections to have different IP addresses. In this way, the MU is enabled to keep the initially assigned IP

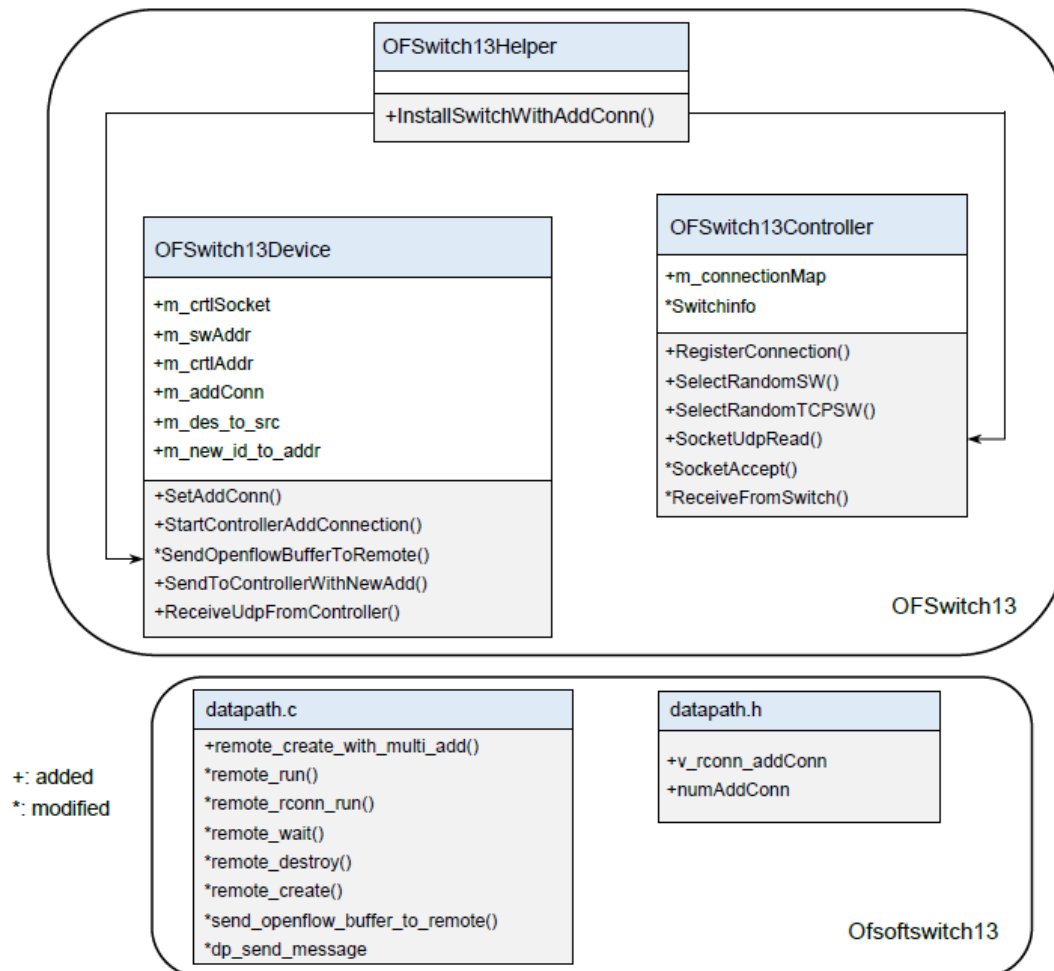


Figure 6.4: Class diagram of the OFSwitch13 module enhancements for multiple DPEs

address despite its location changes and the data packets from service requiring static IP anchoring are directed to initial flow, and where no static anchoring is required, the MU uses the new flow for active communication, while maintaining the reachability for the IP address that is still in use. Consequently, the need for the establishing a tunnel between initial and new DPEs and flow redirection is not required to link different flows, leading in turn to seamless connection mobility during DPEs relocation avoiding a large number of signalling messages. The S+ NC monitor the MU activity, as no traffic is carried in the initial flow (during the timer interval) it starts the releases procedure for the removal of the initial IP address from the IP list. The release cause tag along with the session management and perform the DPE binding information update.

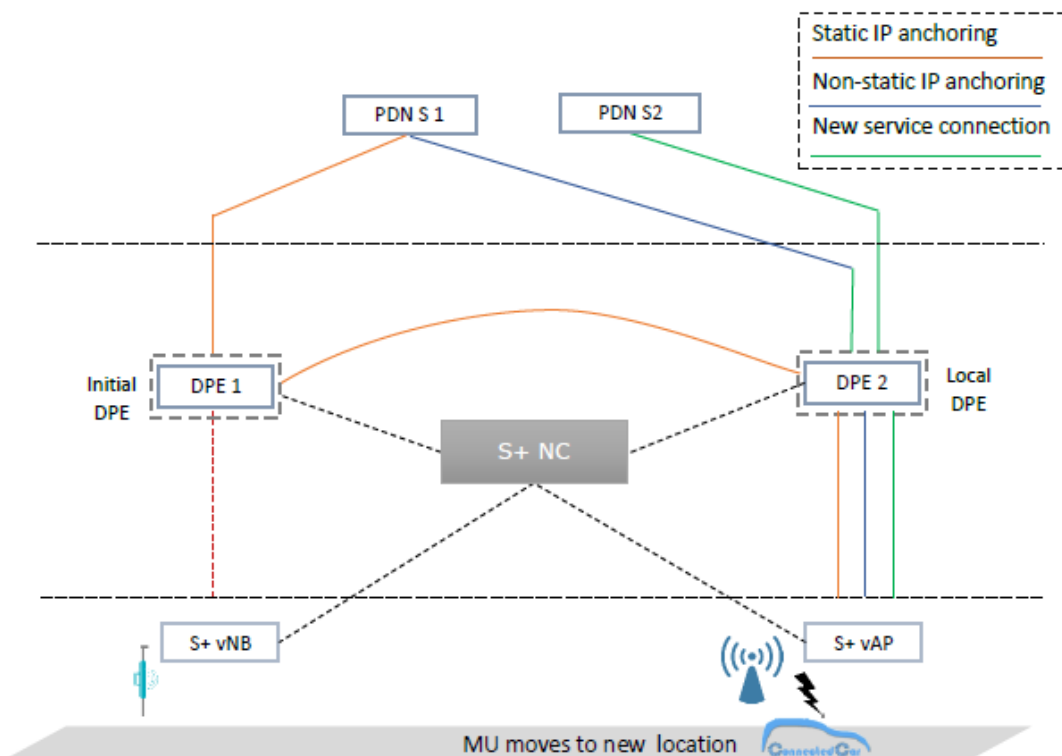


Figure 6.5: Network topology of the evaluation scenario with multiple DPEs.

### 6.2.1 System Procedures

In the following I describe the important call control flows being exchanged across the various implemented S+ MN elements to show how it can be systematically exploited to gain benefits.

### 6.2.2 Initial attachment procedure

A normal attachment procedure was performed illustrating an expression of the proposed architecture. Figure 6.6 shows the signalling diagram of initial attachment, The S+ NodeB receives an attachment request message (1) from MU which includes its subscription profile and intended usage parameters. The S+ NodeB selects the MF and sends AN registration message (2) to MF, where the MF initiates MU authentication and fetches users' subscription profile from S+ DS. After the successful authentication and authorization, an accept indication message (3) is send back to S+ NodeB. The state of MU transitions from deregistered to registered and S+ NodeB reconfigures the radio connection, and forward the attach accept message to the MU. The S+ NodeB triggers the radio access bearer (RAB) setup procedure (4) and also transmits the first OpenFlow Initial Context Setup Request message (3) to AN switch device. Since there is no matching flow entry in the flow table of the AN switch device. The AN switch device triggers an OpenFlow packet-in message (5) to the S+ NC. This message includes some information that is necessary to establish the data plane.

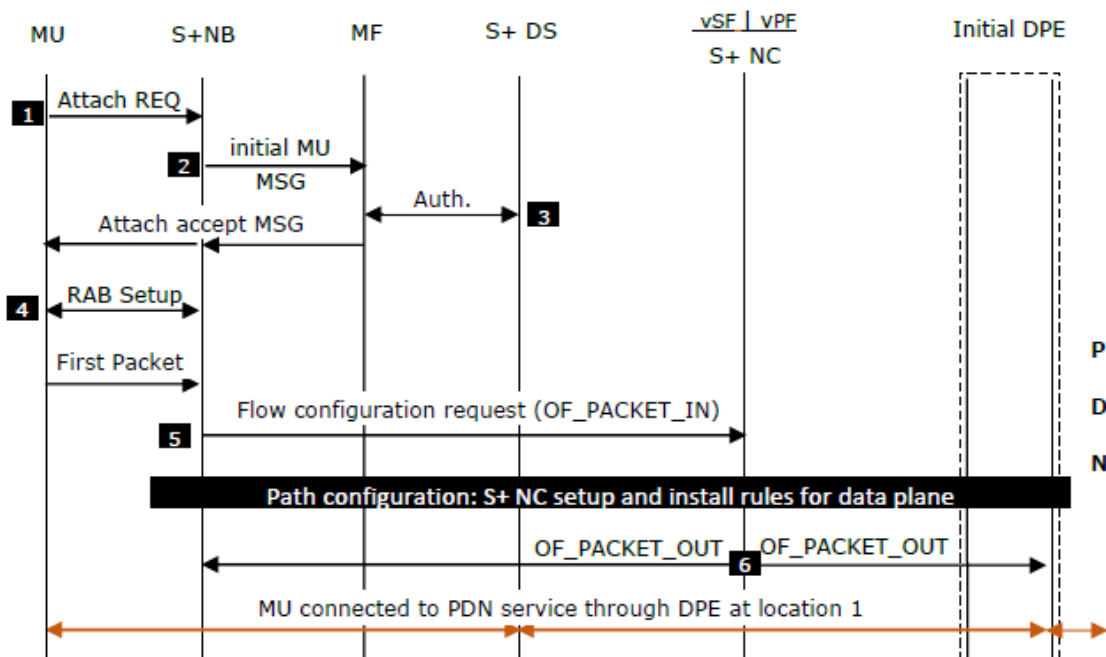


Figure 6.6: Initial attachment operation

The S+ NC analyse the packet header to obtain the session information such as the source and destination nodes IP addresses and collects the transport marking information. S+ NC assigns the DPE by interacting with the SF applications. In turn, the MU obtain an IP address from SF, this also

includes as conferred in [98] creating Binding Cache Entry (BCE) to keep track the MU's location as well as the related information. Then, the S+ NC creates flow rules for subsequent packets that belong to the same session and send them as OpenFlow packet-out message (6) to install them for the S+ NodeB and DPE. In the case where the session requires a specific QoS policy, the S+ NC interact with the SM and PF to get the corresponding policies before downloading the flow rules to the data plane. When the flow rules are associated with a QoS parameter, the S+ NC install them at the DPE. Furthermore, considering the expected handover with MU mobility, The MF in cooperation with the SF notify the needed support for DPE relocation and keep the established connection context. At the end of this procedure, the MU state is transitioned from Idle to connected and a data forwarding path is established from the MU to the correspondent PDN service.

### 6.2.3 Intersystem handover

Here I describe the intersystem handover procedure performed when a MU is in registered and connected states and moves from the coverage area of a source access node (S+ NodeB) to the one of a Target access node (S+ AP) within the S+ AN. As shown in Figure 6.7, the MU is initially connected to a PDN service via DPE 1 and S+ NodeB, The intersystem handover procedure, starts with the handover preparation phase, where based on the pre-defined schedule set up in the simulation, as the MU is placed in positions at a distance from S+ NodeB. A distribution of dwell time is used to trigger the handover and considering the automotive use case, as the handover likely to happens for the users on a highway, I implied the fluid-flow mobility model [110] to derive the average dwell time. I used the free speed distributions model [111] to compute the velocity of the MU. For the sake of simplicity, I assumed that the MU moves in a straight road between the access nodes.

The MF determine the presence of appropriate target access node (in this case a non-3GPP compliant access node S+ AP) and a Handover Request message (2) is send to the target S+ AP. As the targeted S+ AP response with a handover ACK message (3). The source S+ NodeB, in turn, sends a handover required message packaged in an OpenFlow message to the S+ NC, the handover command (4) is send toward the MU by S+ NC and then MU trigger the attachment procedure (5), Access technology specific



procedure for the interworking of the 3GPP and non-3GPP access networks [43]. After attachment, the S+ NC receives the packet-in message (6) from S+ AP and it must connect a set of appropriate DPE that is close to the MU. Together with this procedure, the MU gets a new IP address to be used in the new flow. S+ NC allocates new IP address to MU and has to establish a data forwarding path according to application needs. The S+ NC monitor the MU activity, as no traffic is carried in the initial subflow (during the timer interval) it starts the releases procedure for the removal of an initial IP address from the PDN-S1 IP list. The release cause tag along the connection mobility management and perform the DPE binding information update.

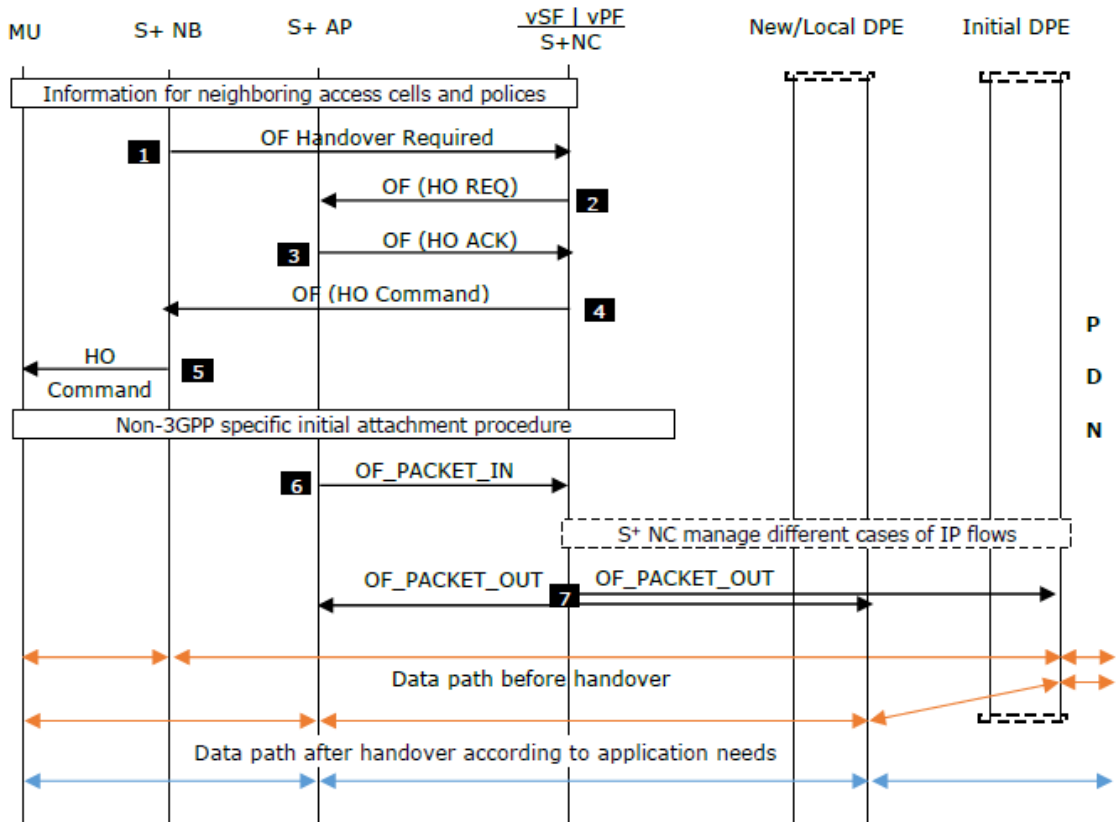


Figure 6.7: Inter-System Handover operation

#### 6.2.4 Qualitative analysis

The ability of validating the functional implementation of the S+ MN architecture has shown that the designed S+ MN architecture is coherently executable and implementable. The pictorial call control flows of system procedures thus created has shown a significant decrease in signalling compared to tunnel and routing based approaches of IP mobility management. Exploiting the open SDN capabilities, the S+ NC is able to parallelize the execution of certain control plane messages thus enabling

the signalling optimisation. The MF in cooperation with the AN notifies the needed support for DPE relocation and the signalling between the AN and S+ NC is not needed for certain cases that are prevalent within the legacy signalling procedure. As shown in Figure 6.5, the S+ NC can manage different cases of IP flows, the data packets from service requiring static IP anchoring are directed to initial flow, so the MU is enabled to keep the initially assigned IP address despite its location changes and where no static anchoring is required, the MU can start a new flow with one of the pre-authorized QoS flow without using control plane signalling. In this way the need for maintaining a tunnel between source and target anchors and flow redirection is not required to link different flows, leading, in turn, to seamlessly connection mobility during IP anchor relocation. The signalling between S+ AN and S+ NC can exceptionally be used when it needs to perform QoS modification, or when the networks want to override the default QoS associated with the flow description. The solution also allows the underlying V2X service to provide additional information to S+ NC application functions, e.g. MU location or potential application locations, connection mobility indicator, etc. In this way, the solution provides an enhanced QoS parameters alignment between application requirements parameters and authorized QoS parameters. Consequently, an efficient data path between MU and V2X application and the re-location of the data plane path is enabled taking into account the different cases of IP flows.

### **6.2.5 Simulation Phase 1 Results and Analysis**

In the following, the gained results from the performed simulation phase 1 are presented and discussed.

#### **6.2.5.1 Intersystem Handover measurements**

I measured, i) The time required by the MU to change from S+ NodeB to S+ AP. ii) The total time consumed by MU in dissociation from the S+ NodeB and becoming ready to route network traffic through S+ AP, that is, the interval between the last ping data packet received or sent by the MU before the handover and the first ping data packet received or sent after the handover. However, it does not consider the duplicate address detection process that should be performed after configuring an IP address on an interface. The measurements have been collected for more than 100 handovers and then plot the average of the runs. For each handover case, I performed the measurement of network traffic at MU's S+ AP interface.

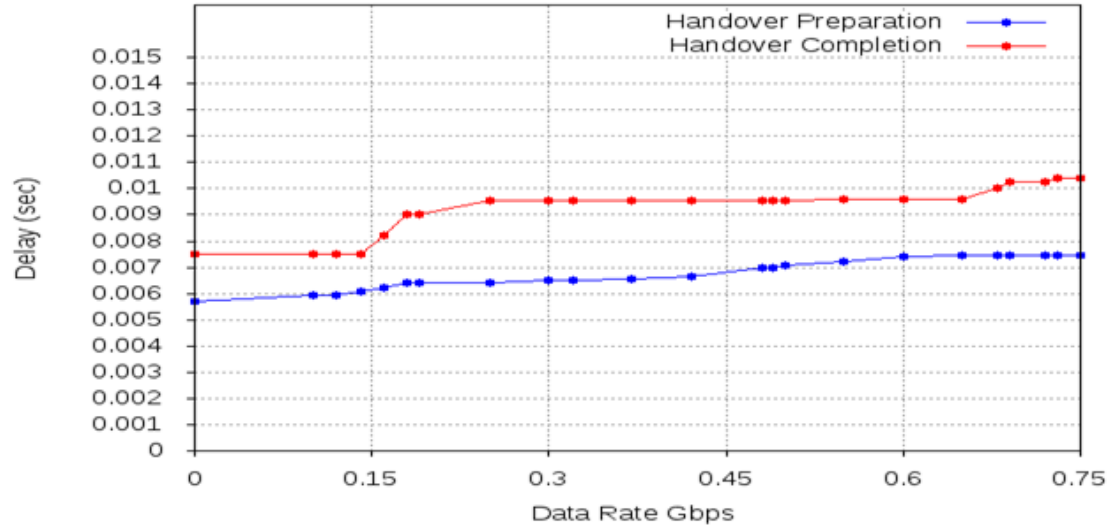


Figure 6.8: Intersystem handover measurements

Figure 6.8 illustrates the intersystem handover preparation and completion times. It is observed that handover preparation and completion times are almost constant for data traffic rates per MU up to 0.75Gbps. The plot indicates that there is a significant reduction in delay performance obtained by the proposed mechanism, whose preparation and completion times are around 5.7 ms and 7.47 ms respectively. In this measurement, the combined impact of QoS aware routing and the distribution of gateway functions is evaluated to investigate the maximum achievable latency reduction. These results fulfil the control plane delay requirements for V2X communication of mobile networks. However, the result is still limited by the minimum processing delay for the packets in the radio network. These delays can be further reduced by different latency reduction techniques of the radio access technology with the complete redesigned to meet the latency targets for 5G.

#### 6.2.5.2 Empirical CDF of Latency of the MU's Downlink data packets

Figure 6.9 presents the Empirical Cumulative Distribution Function (eCDF) of latency values for the first download packets delivered to the moving MU after completion of handover using the proposed solution. The measurements have been collected for more than 100 handovers, where S+ NC manage static and non-static anchoring cases of IP flows as described above. For each measurement, corresponds to each latency value a data rate is also observed as depicted above in Figure 6.8 and 6.9. In presented eCDF plot which is generated from measurement data file using

Gnuplotting script [112] only latency values are depicted. On the horizontal axis the eCDF plots the results of the latency values for first download packets after completing the handover. The fraction of the total number of first download packets after completing handover that are having a latency value less than or equal to a particular value on the horizontal axis is represented on the vertical axis.

The plot on the left indicates that the latency values are lower across 20th percentile level. In non-static anchoring case, here the data forwarding path is through the local DPE that is close to the MU. While the packet delivery time is directly influenced by the time required by data forwarding path in the transport network and a very low latency is observed in this fraction. The latency distribution curves in both cases across 40th percentile is almost overlapping each other suggesting that latency is hardly affected by S+ NC path management according to application in the transport network. The curves towards the right indicate that the latency values are higher across 40th percentile levels. Here, in static anchoring case, the data forwarding path is re-directed from the local DPE to initial DPE to keep the initially assigned IP address despite its location changes. This leads to a path that is longer than the direct one between the MU and its

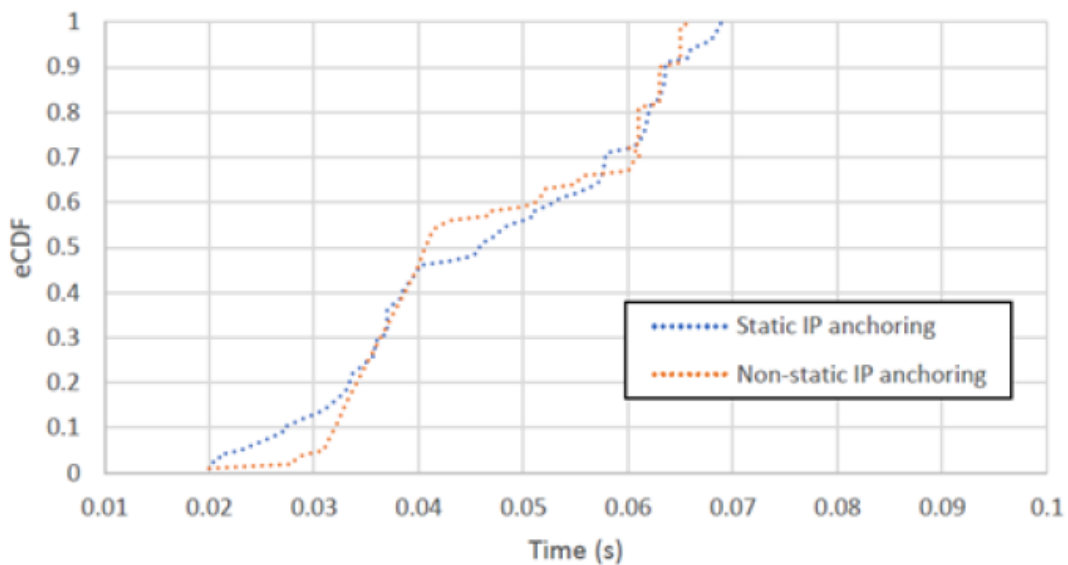


Figure 6.9: Empirical CFD of handover measurement for ping

correspondent service node, adding in turn additional delay in the transport network and a little higher latency is observed in this fraction. The plot also shows that the large fraction of download packet with non-static anchoring performs little faster than one with static anchoring. As mentioned before,

this is because that in this case, a less number of actions must be applied to the incoming flows during the handover procedure and have shorter data path between the MU and its correspondent service node.

For the performance analysis, we defined optimal end-to-end reference latency values between the network elements of the NG-RAN and NG-CN functions in Table 3.

Table 3: Defined Link delay values

	Link Type	Delay Values
1	UE to NG-RAN (3GPP)	1 ms
2	NG-RAN to common CP NF (AMF)	7.5 ms
3	AMF to CP- NF	1 ms
4	AMF to UP-NF	1 ms
5	AMF to AMF	15 ms
6	NG-RAN to UP-NF	7.5 ms
6	UE to AP (non-3GPP AN)	1 ms
7	AP to AMF	7.5 ms
8	S+ AP to UP-NF	1 ms

Corresponding to the simulation setup, I assumed that the related core network CP and UP functions are located at the same geographical location (such as in a data centre). Therefore, the link delay between these network entities is considered to be 1 ms. Only the common CP network function (referred to as AMF using 5G terminology) considered as being geographically distinctly apart, and hence having a 15 ms link delay. The delay between the CP-NF and the AMF is adopted as the maximum 7.5 ms link delay. These values are determined by applying the data presented and implemented in [113] and in [114] respectively.

$$\text{Latency Reduction} = \frac{\text{Delays}_{\text{EPS}} - \text{Delays}_{\text{S Plus}}}{\text{Delays}_{\text{EPS}}} \times 100\% \quad (1)$$

$$\text{Signalling Cost Saving} = \frac{n\text{Msg}_{\text{EPS}} - n\text{Msg}_{\text{S Plus}}}{n\text{Msg}_{\text{EPS}}} \times 100\% \quad (2)$$

For analytical framework, I adopt the formulation presented in [113] for the latency (1) and signalling reduction (2) parameters. where link  $\text{Delays}_{\text{EPS}}$  in (1) refers to the latency values in the EPS network simulation and  $\text{Delay}_{\text{SPLUS}}$  is the latency values obtained in S+ MN simulation. Similarly,  $n\text{Msg}_{\text{EPS}}$

in (2) is the number of messages in the current EPS approach for intersystem handover and  $nMsg_{SPLUS}$  is the number of messages in the proposed approach for intersystem handover. And so, plugging the values from simulation results into (1) and (2). The analytical results obtained for an intersystem handover scenario under consideration, for latency reduction and signalling cost saving parameters have been presented in Table 4. From the results, it is observed that the proposed enhancements to the signalling mechanisms help to reduce the overall intersystem handover signalling compared to current EPS operation. S+ MN enables a significant reduction of CP and UP latencies by up to 50.23 % and 64.66 % respectively, while the signalling cost saving improved of 28.88 %. These obtained performance improvements by analysis restructure aspect followed to implement and validate thus establish the proposed intersystem handover signalling, as well as the proposed S+ MN architecture in enhancing the overall performance.

Table 4: Performance improvement analysis for an Intersystem Handover scenario

Performance parameter	Legacy (EPS) [115]		NG-RAN Ref. Values[48]		S+ MN Realization		Performance Improvements	
	CP	UP	CP	UP	CP	UP	CP	UP
Handover Delay	105 ms	15 ms	50 ms	1 ms	57.5 ms	5.3 ms	50.23 %	64.66 %
Signalling Cost	18		10		11		38.88 %	

### 6.3 Simulation Phase 2: QoS-aware connection mobility with Multipath TCP

To fully appreciate the particular challenges of QoS aware routing during connection mobility, in presented work [116] I proposed multipath TCP to remove the chains of IP preservation of current mobility management solutions. As of the experimental evaluation of proposed solution and to show how it can be systematically exploited to gain benefits, the Linux Kernel implementation of Multipath TCP [81] developed over the last years is recognized as the reference implementation. An overview of the main building blocks and an illustration of the operation of Multipath TCP is described earlier in section 4.4.2.

### 6.3.1 Management of subflows

Figure 6.10 illustrates the architecture of current Multipath TCP kernel implementation to perform management of the subflows. As described earlier in section 4.7.2, it implemented two basic strategies that are referred

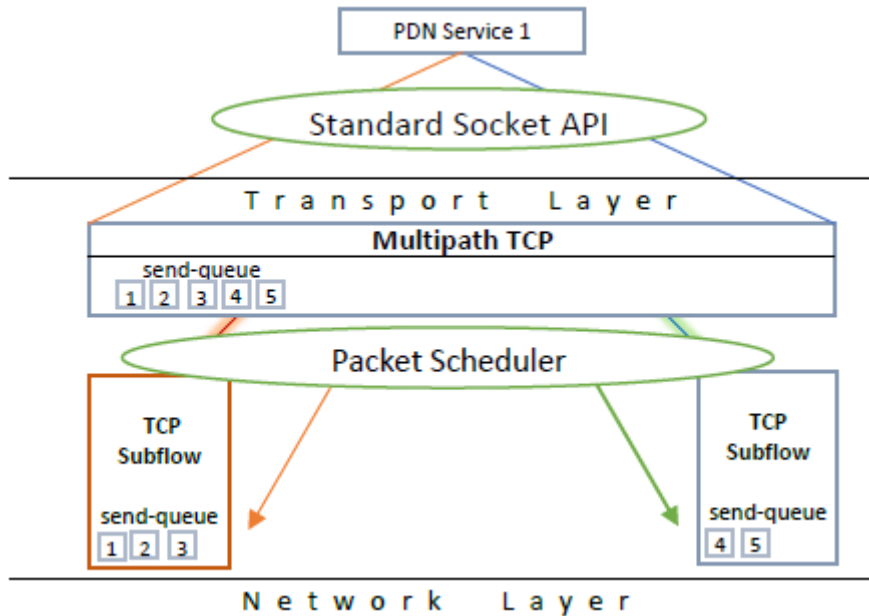


Figure 6.10: The packet scheduler distributes the segments on different TCP subflows

as full-mesh and ndiffports path managers [83]. In addition, a few researchers proposed different approaches for packet scheduling in the multipath nature of networks as surveyed in [117], [118]. In [118] the authors proposed a packet scheduling scheme that favours selecting non-congested subflows to send packets by estimating the available bandwidth and considering the congestion status of subflows. If all subflows are not congested, the subflow with the shortest smoothed RTT is selected. Another scheme referred to as the multipath transmission control scheme (MTCS) attempts to combine congestion control and scheduling to send in-order packets. The scheme uses a load sharing model to distinguish packet loss due to network congestion from the one due to wireless channel condition to find the optimal path. At the same time, it employs a feedback control-based packet scheduling mechanism to maximize the number of data packet sent to the receiver in a timely manner without losing their order. In [119] the authors performs in-order packet scheduling by dispatching packets on each path based on RTT after successfully sending a data packet and receiving an acknowledgement (ACK) for that packet. However, the algorithm uses RTT as an estimate for the forward delay, and it may not be a good estimation due to the asymmetric delay nature of forward and

backward paths. In [117], the performances of several scheduling algorithms are compared, showing that a scheduler minimizing the packet delivery delay yields the best overall performance. Actually, both the scheduler proposed in [119] and the conclusion in [117] agree that the 'fastest' (i.e., largest bandwidth or lowest packet delivery delay) subflow is chosen to send data at any given time. In heterogeneous networks, scheduling data to the subflow based on the lowest round-trip-time (RTT) is beneficial, since it improves the user-experience. It reduces the application delay, which is critical for interactive applications. In other words, the RTT-based scheduler first sends data on the subflow with the lowest RTT estimation, until it has filled its congestion window. Then, data is sent on the subflow with the next higher RTT. In the same way as the round-robin scheduler, as soon as all congestion windows are filled, the scheduling becomes ack-clocked. The acknowledgements on the individual subflows open space in the congestion window, and thus allow the scheduler to transmit data on this subflow.

To accommodate different use cases, a generic and modular path-management the current implementation allows any application to use Multipath TCP. However, most of the code of the existing Multipath TCP implementations is devoted to the transmission and reception of data and the applications are unaware of the multipath nature of the network. As the specific knowledge of an application cannot be known, this is identified as a limitation for applications that could benefit from specific knowledge to use multiple paths according to their needs. A few researchers have explored how Multipath TCP should enable applications to control the utilisation of the different paths. An extension to Multipath TCP has been proposed in [120] that enables to adapt the utilisation of the subflows based on information extracted from the MAC layer. This extension is evaluated experimentally, but there are no details on how it has been implemented. In [108] the author explored how wireless devices can adapt to losses of connectivity. This work also proposed three flow modes of operation (single-path, backup and full-mode) for Multipath TCP on smartphones. In [121] the author proposed some enhancements to the basic socket API that enable applications to add/remove addresses to a Multipath TCP connection. The authors in [122] proposed models of managing the subflows to reduce energy consumption and evaluated the energy impact of using Multipath TCP on smartphones. Recently in [101], the authors implemented a path



manager module above the Linux kernel implementation to enable the applications to interact with the Multipath TCP kernel code over a Netlink [123] interface. The proposed subflow controller was further elaborated with smarter long-lived connections, supports backup subflows and smarter exploitation of flow-based link binding use cases. However, none of the existing Multipath TCP extensions implemented the path management for seamless connection mobility use case.

### 6.3.2 Realization of S+ Path Manager

I extended the Multipath TCP path management to delegates the management of the data paths according to the application needs. In line with design principles defined in section 5.1, we reconsidered the design by clearly separating the Multipath TCP data and control planes. The control plane includes all the functions that manage the subflows that compose a Multipath TCP connection and the data plane includes all functions that deal with the transmission of data. To enable the applications to interact with the Multipath TCP kernel code, we used the Netlink based inter-process communication mechanism [115] supported by the Linux kernel that allows applications to interact with the kernel through messages. This is similar to the approach proposed earlier in [93]. We abstracted Netlink library<sup>1</sup> and linked with the S+ path manager.

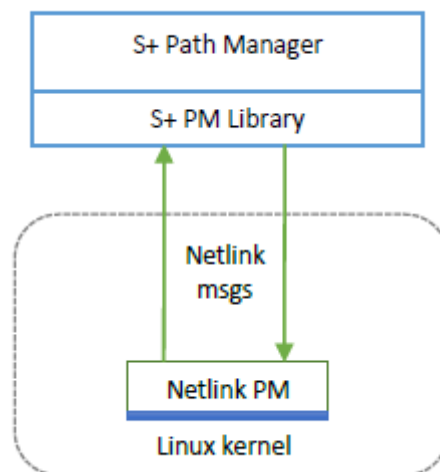


Figure 6.11: The S+ Path management with Netlink support [93].

1: Original author is Gregory Detal, has been ported to a recent kernel version with adaptations to fit with Linux coding style.

As shown in the Figure 6.12 this library running entirely into user space, it interacts with the Netlink path manager that is part of the kernel and enables the S+ path manager to modify the state of Multipath TCP. This S+ path manager uses the existing in-kernel path manager interface (shown in blue in Figure 6.12) and exposes this interface through Netlink.

The simulation consists of three main components, S+ path manager plugin, socket wrapper and gateway manager. These components interact with each other asynchronously and share information via in-memory REDIS database service [124]. The list of S+ path management dependencies is given in Table 5.

*Table 5: S+ path manger dependency*

Parameters	Values
Ubuntu Linux 18.04 LTS	Ubuntu 18.04 LTS
Linux Kernel version	4.19.55.mptcp
Multipath TCP	v0.95
GNU C Compiler	v7.4.0
Embedded Linux Library	v2.1
Argp Library	built-in GNU libc
Linux kernel Multipath TCP user API headers	
Pkg-config	
GNU Compiler Toolchain	e.g. GNU Automake GNU Autoconf GNU Autoconf Archive, GNU Libtool
Redis C API i.e. libhiredis-dev	v0.13
Python version	3.6.7-1
Python Netlink Library	i.e. python3-pyroute2 v0.4.21-0
Python REDIS API i.e.	python3-hiredis v0.2.0-3

The REDIS service plays a vital role in this simulation, it not only stores the current state of all Multipath TCP connections and subflows but also makes it possible to configure future Multipath TCP connections and subflows according to application needs. The first component is a user space S+ path manager plugin written in C language and consists of less than 600 lines of code [125]. It is loaded and set as a default plugin in user space Multipath TCP daemon service. Using the Netlink communication mechanism, it uses Netlink to listens for Multipath TCP events happening in Linux kernel space. Whenever an event e.g. a new MPTCP connection is created or closed, or a subflow for an existing MPTCP connection is added or removed etc. the Linux kernel signals this event to S+ path manager plugin in Multipath TCP Daemon service. The S+ path manager service inspects the event and

update REDIS database accordingly. It ensures that the state of all Multipath TCP connections remains UpToDate in REDIS database so that other components of the simulation can act upon it. The S+ path manager plugin monitors following events.

- A new MPTCP connection is created.
- A new MPTCP connection is established.
- An existing MPTCP connection is closed.
- A new MPTCP subflow is added to a connection.
- A MPTCP subflow is removed from a connection.
- A MPTCP subflow priority has changed.
- A new IP address (IPv4 or v6) is set on a network interface.
- An IP address (IPv4 or IPv6) is removed from the network interface.

At the time of writing, the Multipath TCP daemon service is still in early beta and does not provide any way to modify Multipath TCP connections and subflows. Therefore, Socket Wrapper [126], a separate user space software is used to manage link quality and route priority. Socket wrapper is a shared library that loads before any other library used by a given program and monitors new socket connections from that user space program e.g. vehicular internet, traffic efficiency application etc. Upon detecting new TCP socket connection, it reads user preferences regarding Type of Service (TOS) and QoS parameters for the program in REDIS backend and applies them on socket connection. Thus, TCP packets from the PDN service can be prioritized or deprioritized according to application preferences set in REDIS database. The library also contains example code to manipulate Multipath TCP options on per connection basis. This includes options such as, enable/disable MPTCP support completely, setting and configuring path managers, setting and configuring packet schedulers and management of subflows etc. The code can be easily extended to load these options from REDIS backend and apply them on pre-existing programs that do not already support MPTCP. However, this library requires that the native socket API supports the operations I apply and is backwards compatible. At the time of writing such a support is still under development by Multipath TCP team and is currently unavailable for the public use. The library encapsulates and overrides the whole socket API, so not only I can apply and test socket options but also do just about anything the socket API can do and impose them on any existing programs without having to recompile

them. This can be a great hacking and reverse engineer tool for network communications. A gateway manager component manages network route by allowing MU to set preferred network interface and gateway as default route. It is mainly useful when MU has multiple network interfaces and at least two or more can be active at the same time. The Gateway Manager is a python script that runs in background as a persistent service. It monitors network state via Linux Netlink IPC framework. As soon as network state changes e.g. a down network interface comes up or a new IP address is assigned to it. It reads the REDIS database and loads MU's network preferences. If the current network state does not match MU's preferred setting than it tries to change it accordingly. It deletes the default gateway and sets the preferred gateway to force network traffic to go through it. The preferred network interface and gateway is set in REDIS backend and can be changed on the fly. If Multipath TCP support is enabled on MU, as does in this simulation, then all network communications can be smoothly shifted from non-preferred gateway to preferred gateway without breaking the communication session. The gateway manager can be easily extended to forcefully shutdown non-preferred interface and use only the preferred interface whenever possible. Thus, saving network costs and improving battery life on MUs.

Table 6: Test Parameters

Parmeter name	Description
Number of Network Interfaces	2
Preferred Network interface	500Mbps WIFI 802.11ac
Non-Preferred Network Interface bandwidth	50Mbps 4G LTE
Multipath TCP Path Manager: Netlink	Nelink
Multipath TCP Packet Scheduler	Default
Multipath TCP Sub-Flows	1 per active network interface
Traffic Type	Simple HTTP, HTTPS, iPerf, Video Stream over WebSocket, VoIP call over WebSocket
Number of Concurrently Active Apps: 3	3
Performance Metrics	packet loss, round trip time, delay and average throughput

### 6.3.3 Performance evaluation

I discuss a scenario used to explain the behavior is the intersystem handover of a MU from non-3GPP access network towards a 3GPP access network. The Multipath TCP implementation used in all my evaluation is based on release 0.95. A list of test parameters is given in Table 6. The

Figure 6.12 illustrates the considered handover topology and situation experienced on MU. To create a connection with a PDN service server, the MU sends a SYN segment over the initial access interface (S+ AP). This segment contains the MP\_CAPABLE option that requests the utilisation of Multipath TCP and other information required to identify the connection. The Netlink path manager sends and receives messages that contain information about the connection, the subflow(s), the type of event, etc. The PDN service server replies with a SYN+ACK segment that also contains the MP\_CAPABLE option and a random key. The created event is triggered that contains the four-tuple, the id of the initial subflow.

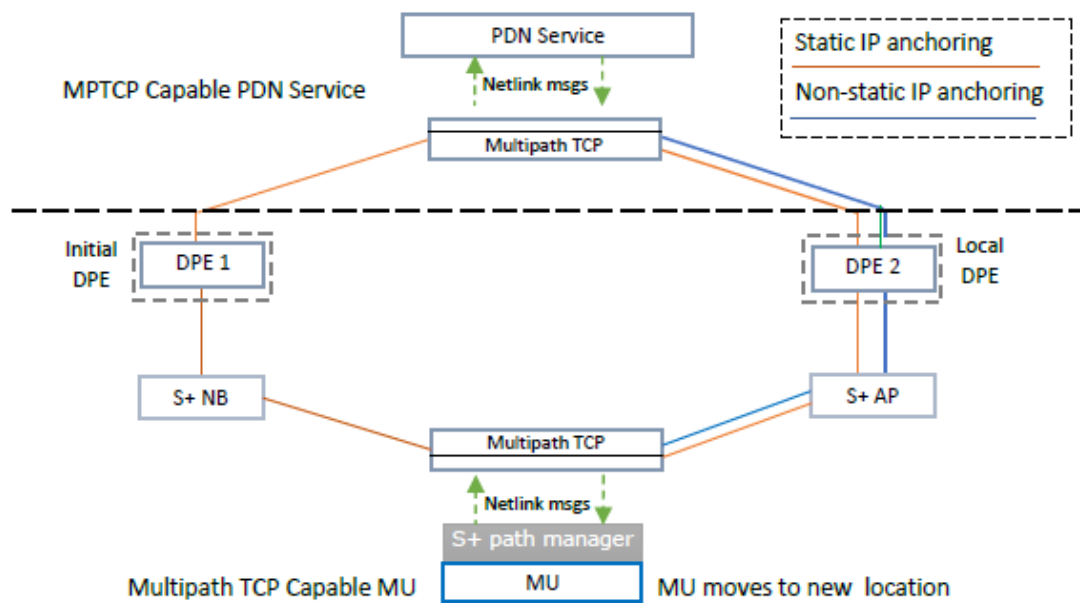


Figure 6.12: The network topology used for simulation

The MU finalises the three-way handshake and the Multipath TCP connection is established. The estab event indicates the success of the three-way handshake. The S+ path manager triggered callback functions specific to received events. At this point, the connection is composed of only one subflow, the one established over the initial interface. Data can be transmitted over this subflow. Moving out of S+ AP coverage area, the radio conditions continues to decrease, and MU need to connect to new access interface. For the handover forecast, I used the link quality as provided by the operating system (Linux provides information about the link quality in linux/include/linux/wireless.h). Figure 6.13 shows the initial access interface (S+ AP) link quality during the experiment. The link quality is a unit-less indicator between 0 and 100, which is determined by the operating system and the device driver using signal strength and noise information. I

manually tuned the forecast decision function to trigger a handover when the link quality continues to decrease after it is less than 30 for three seconds. Link quality threshold of 30 for three seconds is being discussed as an optimal link parameter for triggering a handover in various network research disciplines such as handover triggering in IEEE 802.11 networks [127]. The MU switches towards the 3GPP compliant access interface (S+ NB) and the S+ path manager need to establish a new subflow with Multipath TCP options. To use the new interface, the MU simply sends a SYN segment with the MP\_JOIN option over this interface. This option includes a token derived from the random key exchanged in the MP\_CAPABLE option to identify the Multipath TCP connection to which the subflow must be associated. The PDN service server confirms the establishment of the subflow with a SYN+ACK segment containing the MP\_JOIN option. A new IP address is assigned to the MU and the add\_addr event provide the IP addresses announced by the MU and connecting server. With this event, the S+ path manager can store the addresses of the connecting hosts. In addition to subscribing to some of these events, the implemented library enables the S+ path management to modify the state of Multipath TCP connections through commands. As of this writing, with assumption that the applications wants to obtain the adoptive results with Multipath TCP instead of implying that the application is dumb and only requires a regular byte stream service.

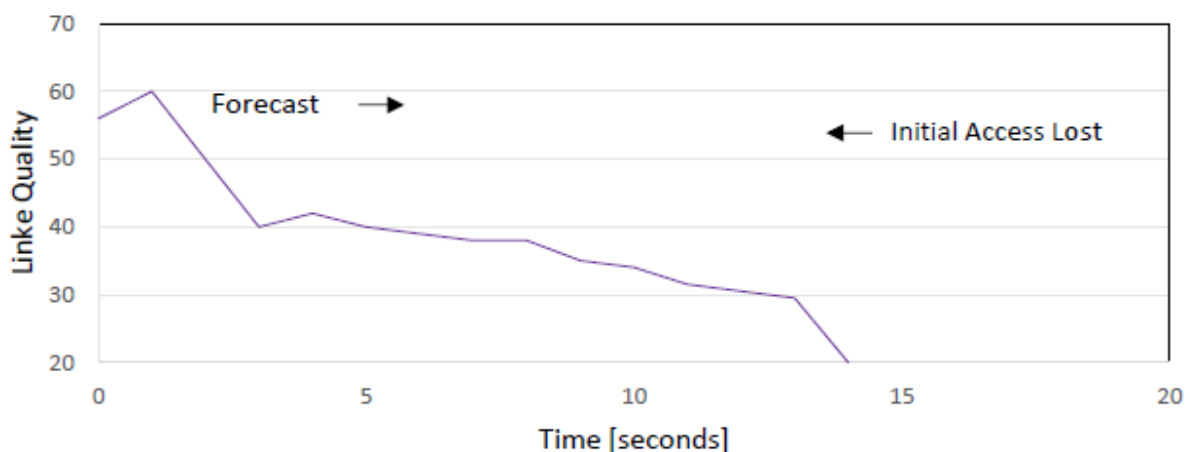


Figure 6.13: Initial access interface (S+ vAP) link quality for handover forecast

As shown in Figures 6.14, 6.15 and 6.16, the implementation supports several types of commands for different cases of IP flows. In full handover

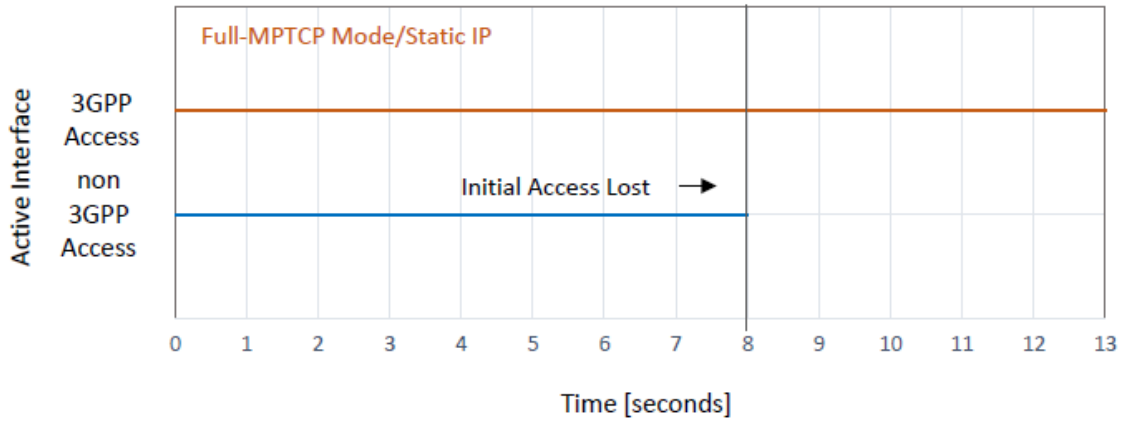


Figure 6.14: Active interfaces for Full-MPTCP Mode

mode, the MU finalises the three-way handshake with an ACK segment. The sub\_estab event is triggered once a new subflow has been established. At this point, the connection is composed of two subflows. all the subflows are used simultaneously between connected MU and server. Figures 6.14 and 6.15 illustrates the active interfaces during the experiment. In Full-Multipath TCP mode (Figure 6.14), MU uses one interface permanently and there is no performance degradation during the handover, but consumes more energy, as both interfaces are active in parallel. In single-path mode (Figure 6.15) the S+ AN connection breaks at second 8, leading to a short downtime.

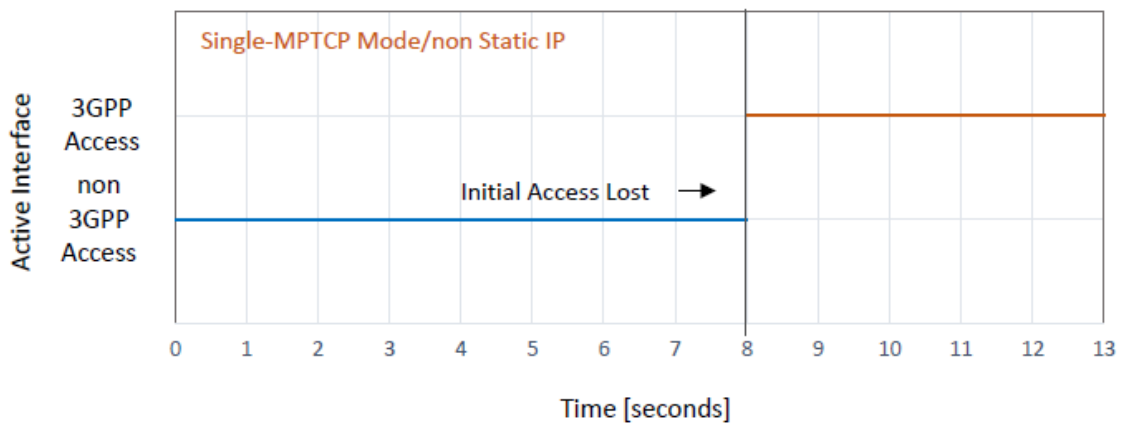


Figure 6.15: Active interfaces for the Single-Path handover at second 8.

Figure 6.16 shows the reproduced results for smooth handover. Moving out of S+ AP coverage area, the given forecast function expects S+ AP access to drop at second 7 and therefore triggers the handover to S+ NB. When the actual S+ AP access is lost at second 15, the MU access is already fully established. Thus, the proposed approach allows a smooth handover with only 8 seconds of parallel active interfaces. The data packets from service

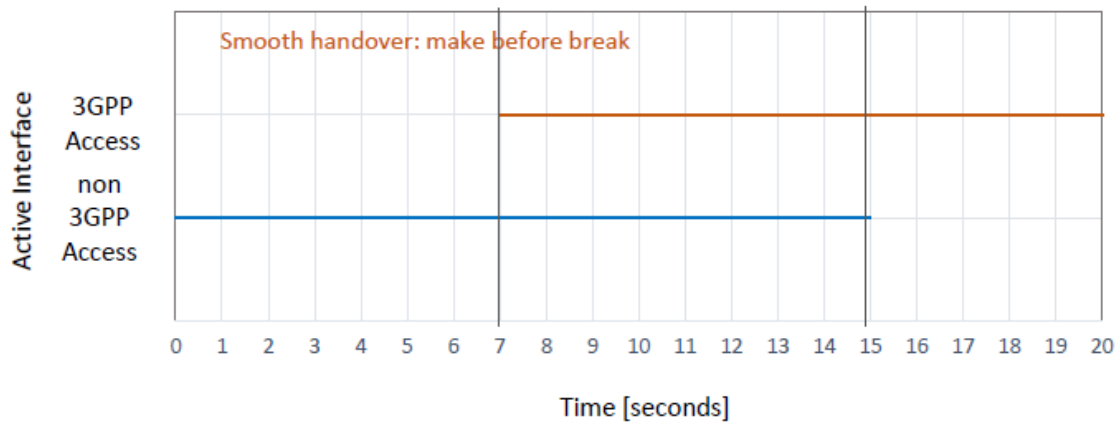


Figure 6.16: smooth handover using the S+ AP link quality for handover forecast.

requiring static IP anchoring are directed to initial subflow, so the MU is enabled to keep the initially assigned IP address despite its location changes and where no static anchoring is required, the MU uses the new subflow for active communication, while maintaining the reachability for the IP address that is still in use. In this way the need for the maintaining a tunnel between source and target anchors and flow redirection is not required to link different subflows leading in turn to seamlessly connection mobility during interface relocation avoiding a large number of signalling messages. The S+ path management monitors the MU activity, as no traffic is carried in the initial subflow (during the timer interval) it starts the releases procedure for the removal of initial IP address from the IP list. A corresponding command to `rem_addr` event allows to remove any established subflow, provided the IP addresses removed by the PDN service server with Multipath TCP options. This is more flexible than the existing in-kernel path managers that immediately create/close subflows. This also reduces the state maintained for each Multipath TCP connection in the kernel. The controller can also retrieve information from the control block of the Multipath TCP connection or one of the subflows. In practice, this is equivalent to the utilisation of the `TCP_INFO` socket option on Linux. Thanks to these commands with a combination of REDIS service, Socket Wrapper, Netlink capabilities and MPTCP options be used to provide QoS-aware routing. With implied MPTCP connection of MPTCP capable MU and PDN-Service and will be able to synchronize the MU traffic using different IP addresses. A user space S+ Path management can therefore control the MPTCP subflows and the policy to apply over those new subflows for every connection.



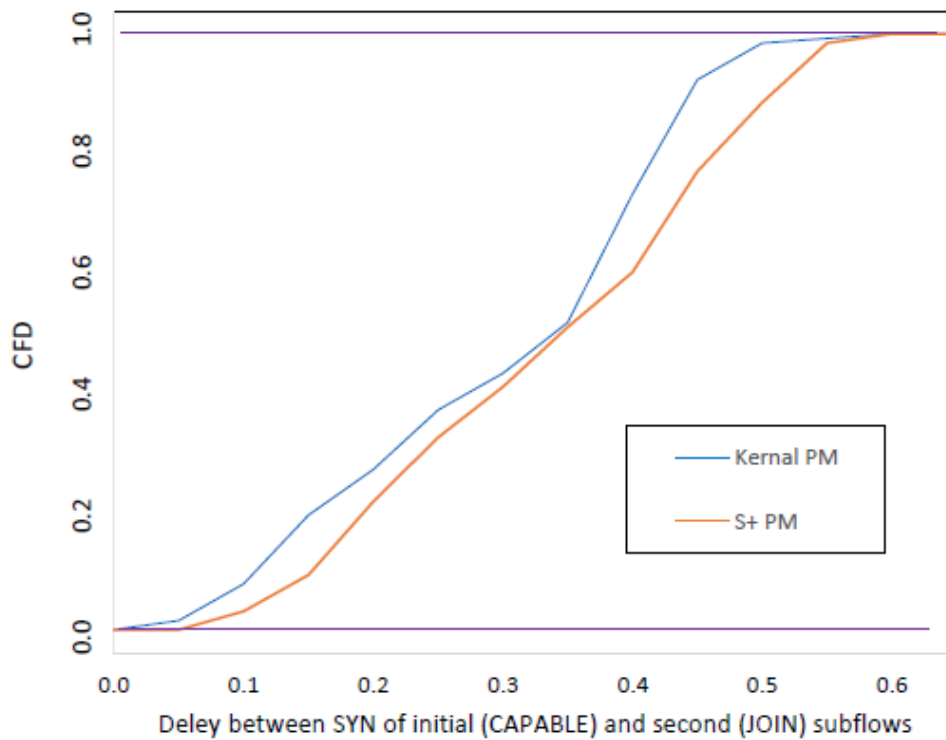


Figure 6.17: Comparison of Kernel path manager and S+ PM

I also reproduced the results by current MPTCP implementation kernel path manager to then compare them with the proposed S+ PM. Figure 6.17 shows the reproduced results for the delay between the SYN of the initial subflow (i.e., containing the MP\_CAPABLE option) and the SYN of the second subflow (i.e., containing the MP\_JOIN option). It shows that the in-kernel path manager is slightly faster than the S+ PM. This additional delay is small and remains acceptable for a MU.

## 7 Conclusions

In this thesis, an SDN plus Virtualization featured Mobile Network (S+ MN) architecture is designed and taken up as baseline reference model, aiming at the further improvements to support the access requirements for diverse user groups. S+ MN architecture through separation of data and control planes provides high flexible architecture, allows the efficient and adaptive sharing of network resources, enable the convergence of multiple heterogeneous networks through open and technology independent interfaces. An overview of state-of-the-art mobile networks architecture has been presented in chapter 1. The protocol stack across the different interfaces is explained, along with an overview of the functions provided by the different protocol layers. It has been seen that 5G system architecture has the transition from a network of entities to a network of functions. There are many dimensions and novel technologies included in 5G system that set new challenges for the new research advances. The open issues, which still needs to be addressed while implementing the 5G system architecture are identified and discussed at the end of chapter 3. Based on this and further analysis on the architecture details have led to the basic design selections in chapter 4 for the proposed enhancements. The common characteristics of the automotive web of services and specific requirements that an optimized communication system is expected to fulfil are framed. Furthermore, the scenarios addressed by the European ITS communication architecture from the main fields of applications for connected vehicle are discussed. It has been concluded that the main challenge comes of today's mobile networks is that multiple services are supported over the same architecture, which could not natively support specified V2X communications. By evaluating different potential technologies, it has been established that the network slicing is the most relevant building block to create customized mobile network for specific user group. To this effect, the S+ MN architecture has first been introduced on a high level in chapter 5. I. The Key principles of S+ MN architecture are defined that can be exploited to gain benefits in multiple use cases. In line with these directions, the enhanced behavior of S+ MN architecture for the abstraction and sharing of network resources is introduced. An S+ slicing scheme is designed to be generic and adaptable to specific user class. The identified specific issues that needs to be addressed while applying the proposed slicing scheme are discussed in section 5.5. Next, an automotive slice is instantiated that

includes customized access as well as core, and transport network functions based on V2X service requirements. As a way to obtain the fine-grained access services, the potential functional grouping provided by the S+ MN architecture has been introduced. Through the design and placement of specific access and core network functions in respective set of automotive slices, the individual and personalized QoS is preempted for a specific automotive use case category. Moreover, the task of QoS provisioning is approached from the perspective of information centric with the input from service requirements. In this way, the solution provided an enhanced QoS parameters alignment between application requirements parameters and authorized QoS parameters. The proposed slicing architecture has been further elaborated through a number of exemplary procedures. The pictorial call control flows of system procedures shown a significant decrease in signaling messages processed by control entities relative to current mobile network architecture. The proposed slicing scheme also provided the capability of flexible mechanism for slice instance selection, management and orchestration.

In chapter 6, the thesis applied an experimental design to the evaluation of the proposed enhancements. In an implementation phase 1, a functional setup validates the designed S+MN architecture including its core functionality implemented using the ns-3 network simulator. Distribution of gateway functions is implemented to solve the problem of unnecessary long routes and delay. The data gathering is realised with S+ DS to enable the data gathering on the access level to collect the context data of the MU as well as of the S+ MN state and utilisation. Exploiting the open SDN capabilities, the re-location of data plane path is performed by S+ NC taking into account the different cases of IP flows. For intersystem handover case, I performed the measurements of network traffic, handover preparation, delay and total execution times. The results fulfilled the control plane delay requirements of the 5G networks. I then specified an analytical framework and the reference optimal performance parameters. Utilizing this framework, I have performed a performance improvement analysis based on latency reduction and signalling cost saving. The results showed that the proposed enhancements help to reduce the overall intersystem handover latency and signalling cost compared to the current EPS operation.

Furthermore, I experimentally prove the feasibility of using Multipath TCP for connection mobility in intersystem handover scenario. The experiments run over the Linux Kernel implementation of Multipath TCP developed over the last years. I extended the Multipath TCP path management to delegates the management of the data paths according to the application needs. I analyze Multipath TCP's performance in various flow modes and find that it enables connection mobility offering an efficient data path between MU and V2X application. The results showed that the proposed Multipath TCP enabled the efficient relocation of data path taking into account different cases of IP flows and removed the chains of IP address preservation for seamless connection mobility during intersystem handover.

A number of papers covering different aspects of the proposed S+ MN architecture, functionalities, and achieved results in this research have been presented and published in refereed journal and conferences. A list of the published outputs can be found in the Appendix A. I believe that this work will pave the path towards the customized vehicular communications system.

## **7.1 Achievements of the Research**

To summarize, the main contributions of this work can be stated as follows:

- Realization of the key principles of upcoming Fifth Generation (5G) of the mobile network as an S+ MN Architecture. The control plane functions are decoupled from the data plane functions allowing independent scalability and evolution. The control path is extended down to the access and user nodes to enable independent access and data session handling. Various types of user-centric, network-centric and context-centric data is collected and managed in unified data server and use the additional information from big data analytics within the system to optimize QoS and Mobility control and management.
- Suggested automotive slicing solution, elaborate the partition(s) of the core network and the radio access network resources, as well as configuration of the automotive slice instances, to support different V2X use cases. A Mobile Network slicing scheme is designed to be generic and adaptable to specific user class. Configuration and the control flows for how to instantiate and control an Automotive Slice

Instance during its life cycle. which provides certain network characteristics (e.g. low latency, individual traffic steering, QoS, Mobility management, Service and/or Session continuity etc.).

- Improvement for the resource management, QoS and Mobility control and management in automotive slicing scenario.
- Development of a functional validation setup to show the working of S+ MN architecture.
- Description of the procedures of current mobile networks to S+ MN context using the proposed S+ NC; Initial attachment, service setup, intersystem handover, the data plane management etc., and avoiding a large number of signalling messages.
- Distribution of gateway functions is followed and (Re)selection of efficient data plane paths is enabled. By analysing current research of IP mobility management in SDN based mobile network architecture; the proposed architecture implied upper layer transport protocols functions to remove the chains of IP address preservation for session continuity during IP anchor relocation without the use of bi-directional tunnels between the initial and new router.
- Application oriented forwarding of IP traffic through the use of distributed IP anchors and select the right one for use in the following cases. i) The MU acts as a server and requires a static IP address for incoming IP flows. Static anchoring at the initial router will be required. ii) No fixed IP address, i.e. the MU acts as client, but IP session continuity: No static anchoring at the Initial router will be required. iii) No fixed IP address and no IP session continuity: no static anchoring at the Initial router will be required.

## **7.2 Limitations of the research**

Despite having met the research objectives stated above, a number of limitations can be identified, which were caused by current state of practical and infrastructural access reasons. The key limitations of the research are summarised as follows.

1. The seamless connection mobility during an intersystem handover scenario is investigated using a single S+ NC. To address scalability in a dynamically changing network condition, the selection of an appropriate AN could be explored with the cooperation of multiple

distributed S+ MN controller. Furthermore, there are several models of SDN such as OpenSDN (using OpenFlow protocol), SDN API model, and SDN via VLAN tunnels [128], this work has realised the S+ MN architecture only using open SDN capabilities.

2. To provide QoS according to application needs, this work proposed and implemented the use of multiple simultaneous connections by connected MUs. A repercussion of this approach is that MUs need to consume extra energy and resources to preserve the state for the established TCP connections. Although the impact of energy on considered automotive use case is uncritical in comparison with other mobile devices. In order to alleviate the limitation, more efficient provisioning for distribution of gateway functions will be required, incorporating mechanisms of controlled resource consumption in all mobile network elements that are placed on the data forwarding path. Furthermore, this work has not investigated much on the scalability and effectiveness of QoE-aware collaborative service management.
3. The functional validation setup of S+ MN architecture was restricted to the basic principle, implement as much functionality as necessary to examine the proposed architecture and its functionalities. The simulation implementation considers only the signalling cost and latency related parameters. Furthermore, the realization of S+ path manager to delegates the management of the data paths according to the application needs is implemented using multiple components, the S+ path manager plugin, socket wrapper and gateway manager. These components interact with each other asynchronously and share information via in-memory REDIS database service. Instead of directly controlling the MPTCP connections, separate user space software (Socket Wrapper) is used to manage link quality and route priority. The reason for this approach was the current state of baseline MPTCP kernel implementation and limited availability of API daemon service, the Multipath TCP daemon service was in early beta at the time of implementation and did not provide any way to modify Multipath TCP connections and subflows.

### 7.3 Suggested scope for future work

The research presented by this thesis has advanced the field of QoS-aware connection mobility in SDN and virtualization featured mobile networking. However, there are a number of areas in which future work could be carried out to further advance upon the findings of this research. The details of future work are listed as follows.

1. In this thesis, we suggested four well-defined automotive slices categories (section 5.6). Even though, it is still not trivial how to classify a particular automotive application into these four categories. For instance, an application may require session control functionalities and low-latency V2V and V2P periodic or event driven messages at the same time, which generates a particular requirement for an extra network slice. Therefore, further work could be done for the granularity of automotive slicing formation in terms of the promised performance.
2. During inter-slice handover, a MU state may spread over multiple slice instances. To obtain MU state from one virtual entity to the other or pre-empt the MU state handling priority require extra procedure in the core network. Additional inter-slice inter-networking procedures could be realised such as with steady state, random movement pattern and variable MU speeds. Furthermore, in order to support new advance services of future automotive applications, new functions may be needed besides proposed functional groupings. Therefore, it has to be further identified what are the new required functions, messages, procedures and where the S+ MN architecture can be adapted to bring benefit to new use cases and application areas.
3. At the time of implementation of S+ Path Manager, the Multipath TCP daemon service was in early beta state and did not provide any way to modify Multipath TCP connections. Therefore, a separate user space software is used to manage link quality and route priority. With the availability of APIs, a more optimal and direct approach could be used to provide QoS-aware routing. Further ns-3 implementation work could be done using SDN API model. These new possibilities should be addressed in future research.

## References

- [1] Cisco, "Cisco Visual Networking Index: Forecast and Trends." [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. [Accessed: 09-Mar-2020].
- [2] Qualcomm Driving Network Evolution, "The 1000x Data Challenge." [Online]. Available: <https://www.qualcomm.com/documents/1000x-mobile-data-challenge>.
- [3] Deintermixture, "When Television Almost Became All-UHF." [Online]. Available: <http://www.uhftelevision.com/articles/deintermix.html>.
- [4] W. Webb, "On using white space spectrum," *IEEE Commun. Mag.*, vol. Volume: 50, no. Issue: 8, pp. 145 – 15, 2012.
- [5] G. B. Dawood M, Fuhrmann W, "Assay of White Space Technology Standards for Vehicular Cognitive Access," in *Tenth International Network Conference, INC 2014*, 2014, pp. pp23-33.
- [6] 3GPP, "Scope and Objectives," 2017. [Online]. Available: [https://www.3gpp.org/ftp/Inbox/2008\\_web\\_files/3GPP\\_Scopeand310807.pdf](https://www.3gpp.org/ftp/Inbox/2008_web_files/3GPP_Scopeand310807.pdf).
- [7] IEEE, "802.11 Timelines." [Online]. Available: [http://www.ieee802.org/11/Reports/802.11\\_Timelines.htm](http://www.ieee802.org/11/Reports/802.11_Timelines.htm).
- [8] 3GPP TS 23.799, "Study on Architecture for Next Generation System." Release 15, v. 14.0.0.
- [9] R. Tafazolli, "ETSI Future Mobile Summit, Why 5G?"
- [10] SDN Architecture, "Issue 1, Open Networking Foundation," 2014. [Online]. Available: [https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR\\_SDN\\_ARCH\\_1.0\\_06062014.pdf](https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf).
- [11] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Transactions on Network and Service Management*. 2012.
- [12] 3GPP TS 23.501, "System Architecture for the 5G System;" Release 15, v. 15.0.0, December, 2018.
- [13] Highlights of 2017 Telecom Infrastructure Project., "[online] URL:" [Online]. Available: <https://techblog.comsoc.org/category/c-ran/>.
- [14] J. Lee, J. K. Han, and J. Zhang, "MIMO technologies in 3GPP LTE and LTE-advanced," *Eurasip J. Wirel. Commun. Netw.*, 2009.
- [15] P. Tracy, "What is mm wave and how does it fit into 5G." .
- [16] P. C. GROUP, "Connected Mobility Global Forecast 2016," 2017.
- [17] H. J. Einsiedler, A. Gavras, P. Sellstedt, R. Aguiar, R. Trivisonno, and D. Lavaux, "System design for 5G converged networks," in *2015 European Conference on Networks and Communications, EuCNC 2015*, 2015.
- [18] 3GPP TS 23.303, "Proximity-based services (ProSe);," Release 15, v." 15.0.0, June 2017., 2017.
- [19] ETSI White Paper No. 11 *et al.*, "Mobile edge computing—A key technology towards 5G. ETSI White Paper," *IEEE Internet Things J.*, 2018.
- [20] NGMN, *Description of Network Slicing Concept by NGMN Alliance*. 2016.
- [21] X. Li *et al.*, "Network Slicing for 5G: Challenges and Opportunities," *IEEE Internet Comput.*, 2017.
- [22] F. Z. Yousaf *et al.*, "Network slicing with flexible mobility and QoS/QoE support for 5G Networks," in *2017 IEEE International Conference on Communications Workshops, ICC Workshops 2017*, 2017.
- [23] 3GPP, "About 3GPP." [Online]. Available: <https://www.3gpp.org/about-3gpp>.
- [24] T. I. of E. and E. Engineers, "https://supportcenter.ieee.org/app/answers/detail/a\_id/190/~/-what-is-ieee%3F." .
- [25] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access." Release 15, v. 15.0.0, June, 2017.
- [26] H. Holma and A. Toskala, *LTE for UMTS Evolution to LTE-Advanced*. 2011.
- [27] L. Liang, W. Cheng, W. Zhang, and H. Zhang, "Orthogonal Frequency and Mode Division Multiplexing for Wireless Communications," in *2018 IEEE Global Communications Conference*,



- GLOBECOM 2018 - *Proceedings*, 2018.
- [28] 3GPP TS 24.301, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)." Stage 3.
  - [29] 3GPP TS 23.203, "Policy and charging control architecture." Release 15, V15.5.0, June 2019.
  - [30] 3GPP TS 24.312, "Access Network Discovery and Selection Function (ANDSF) Management Object (MO)." Release 15, V15.5.0, June 2019.
  - [31] 3GPP TR 38.802, "Study on new radio (NR) access technology physical layer aspects." March, 2017.
  - [32] 3GPP TS 38.300, "3rd Generation Partnership Project, NR and NG-RAN overall description." v.15.2.0, Stage 2.
  - [33] ETSI TS 138 300, "5G NR Overall description." version 15.3.1, Release 15, Stage 2, October 2018.
  - [34] Keysight Technologies, "5G Waveform Evaluations for mmWave Communication Using SystemVue." Application note, 5G PPP mMAGIC project, European Commission H2020 program, 2019.
  - [35] Keysight Technologies', "Simulation for 5G New Radio System Design and Verification." White Paper: Published in USA, 5992-3032EN, 2018.
  - [36] 3GPP TS 29.518, "5G System; Access and Mobility Management Services." Version 15.1.0, Release 15, Stage 3, October 2018.
  - [37] Martin Sauter, *Communication systems for the mobile information society*. John Wiley & Sons. p. 131. ISBN 978-0-470-02676-2., 2006.
  - [38] ETSI TS 138 413, "NG-RAN; NG Application Protocol (NGAP)." version 15.0.0, Release 15, July 2018.
  - [39] R. Stewart, "Stream Control Transmission Protocol PROPOSED STANDARD," *RFC 4960*, 2007.
  - [40] 3GPP TS 38.413, "NG-RAN; NG Application Protocol (NGAP)." Release 15, v. 15.0.0.
  - [41] Frederic Firmin, "Non-Access Stratum (NAS) and its protocols EMM and ESM." [Online]. Available: <https://www.3gpp.org/technologies/keywords-acronyms/96-nas>.
  - [42] 3GPP TS 24.501, "Non-Access-Stratum (NAS) protocol for 5G System (5GS)." version 16.0.02, Release 16 Stage 3, March 2019.
  - [43] 3GPP TS 24.502, "Access to the 5G System (5GS) via non-3GPP access networks." Stage 3, Version 15.0.0, Release 15, September 2018.
  - [44] 3GPP TS 36.300, "E-UTRA and E-UTRAN; Overall description." Release 15, v. 15.1.0, Stage 2, October 2018.
  - [45] 3GPP TS 37.324, "E-UTRA and NR; Service Data Adaptation Protocol (SDAP) specification." V15.1.0, 2019.
  - [46] 3GPP TS 23.502, "Procedures for the 5G System (5GS)." Release 16, V16.2.0, September 2019.
  - [47] 3GPP TS 23.501, "System Architecture for the 5G System." Stage 2; Release 15, v. 15.6.0, June, 2019.
  - [48] International Telecommunications Union, "ITU towards 'IMT for 2020 and beyond,'" *www.itu.int*, 2016. .
  - [49] W. Fan, P. Kyosti, M. Rumney, X. Chen, and G. F. Pedersen, "Over-the-air radiated testing of millimeter-wave beam-steerable devices in a cost-effective measurement setup," *IEEE Commun. Mag.*, 2018.
  - [50] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, 2017.
  - [51] P. Zhang, X. Yang, J. Chen, and Y. Huang, "A survey of testing for 5G: Solutions, opportunities, and challenges," *China Communications*. 2019.
  - [52] Keysight, "The Challenge of the First Commercial 5G Service Deployment," 2019.
  - [53] E. J. Oughton and Z. Frias, "The cost, coverage and rollout implications of 5G infrastructure in Britain," *Telecomm. Policy*, 2018.
  - [54] F. ranco C. Marco Chiani, E. Paolini, *Open issues and beyond 5G*. CNIT/University of Bologna Italy.
  - [55] A. Morgado, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "A survey of 5G technologies:

- regulatory, standardization and industrial perspectives," *Digit. Commun. Networks*, 2018.
- [56] H. Zhang, C. Jiang, J. Cheng, and V. C. M. Leung, "Cooperative interference mitigation and handover management for heterogeneous cloud small cell networks," *IEEE Wirel. Commun.*, 2015.
  - [57] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-Enabled Tactile Internet," *IEEE J. Sel. Areas Commun.*, 2016.
  - [58] B. W. Kolosz and S. M. Grant-Muller, "Appraisal and evaluation of interurban ITS: A european survey," *IEEE Trans. Intell. Transp. Syst.*, 2015.
  - [59] 3GPP TR 22.885, "Study on LTE Support of Vehicle to Everything (V2X) Services." Release 14, v. 14.1.0, Dec. 2015.
  - [60] 3GPP TR 22.886, "Study on Enhancement of 3GPP Support to 5G V2X Services." Release 15, v. 15.2.0, June 2018.
  - [61] ETSI EN 302 665, "Intelligent Transport Systems (ITS); Communications Architecture." 2010.
  - [62] ETSI TS 102 637, "ITS; Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service." 2011.
  - [63] C. D. and C. F. S. Neumeier, N. Gay, "On the Way to Autonomous Vehicles Teleoperated Driving." AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium, Dortmund, Germany, pp. 1-6, 2018.
  - [64] 3GPP, "Service Requirements for Enhanced V2X Scenarios," 2019. [Online]. Available: [www.3gpp.org/DynaReport/22186.htm](http://www.3gpp.org/DynaReport/22186.htm). [Accessed: 18-Jan-2019].
  - [65] O. Altintasi, H. Tuydes-Yaman, and K. Tuncay, "Detection of urban traffic patterns from Floating Car Data (FCD)," in *Transportation Research Procedia*, 2017.
  - [66] 3GPP TR 38.913, "Study on Scenarios and Requirements for Next Generation Access Technologies." V15.0.0 Release 15, June 2018.
  - [67] 3GPP TR 22.186, "Technical specification group services and system aspects. Enhancement of 3GPP support for V2X scenarios." Release 15. Sept. 2018.
  - [68] 3GPP TR 22.891, "3GPP, "Technical Specification Group Services and System Aspects; Feasibility Study on New Services and Markets Technology Enablers." Release 14, v. 14.2.0, Sept. 2016.
  - [69] E. K. Lee, M. Gerla, G. Pau, U. Lee, and J. H. Lim, "Internet of Vehicles: From intelligent grid to autonomous cars and vehicular fogs," *Int. J. Distrib. Sens. Networks*, 2016.
  - [70] I. F. Mohamed Zain, A. Awang, and A. Laouiti, "Hybrid MAC protocols in VANET: A survey," *Adv. Intell. Syst. Comput.*, 2017.
  - [71] R. Wang, H. Hu, and X. Yang, "Potentials and challenges of C-RAN supporting multi-RATs toward 5G mobile networks," *IEEE Access*, 2014.
  - [72] O. Khattab and O. Alani, "An overview of interworking architectures in heterogeneous wireless networks: Objectives, features and challenges," in *Proceedings of the 10th International Network Conference, INC 2014*, 2014.
  - [73] R. 3GPP TS 23.402, "Architecture enhancements for non-3GPP accesses." Release 15, v. 15.0.0, December 2017.
  - [74] O. Alliance, "OMA Device management protocol," *Draft Version*, 2006.
  - [75] E. Piri and K. Pentikousis, "IEEE 802.21: Media Independent Handover Services," *Internet Protoc. J.*, 2009.
  - [76] H. Ameer, M. Esseghir, L. Khoukhi, and L. Merghem-Boulahia, "Enhanced MIH (media independent handover) for collaborative green wireless communications," *Int. J. Commun. Syst.*, 2017.
  - [77] W. Fuhrmann, M. Dawood, and B. V. Ghita, "Prospects for TV White Space access in the Evolved Packet Core environment," in *2015 Internet Technologies and Applications, ITA 2015 - Proceedings of the 6th International Conference*, 2015.
  - [78] J. C. Zuniga, C. J. Bernardos, A. De La Oliva, T. Melia, R. Costa, and A. Reznik, "Distributed mobility management: A standards landscape," *IEEE Commun. Mag.*, 2013.
  - [79] F. Giust, L. Cominardi, and C. Bernardos, "Distributed mobility management for future 5G

- networks: Overview and analysis of existing approaches," *IEEE Commun. Mag.*, 2015.
- [80] M. Dawood, W. Fuhrmann, and B. V Ghita, "Improvements for DMM in SDN and Virtualization Based Mobile Network Architecture," *Int. J. Mob. Netw. Commun. Telemat.*, 2018.
  - [81] I. Draft, *RFC 6824: TCP Extensions for Multipath Operation with Multiple Addresses*. 2015.
  - [82] I. Draft, "TCP Extensions for Multipath Operation with Multiple Addresses V0," *PhD Propos.*, 2013.
  - [83] K. Wang *et al.*, "On the path management of multi-path TCP in internet scenarios based on the NorNet testbed," in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2017.
  - [84] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services," *IEEE Wirel. Commun.*, 2017.
  - [85] ONF TS-025, "OpenFlow Switch Specification 1.5.1," *Current*, 2015.
  - [86] ONF, "Wireless and Mobile Working Group." [online]: <https://www.opennetworking.org/tag/wireless-and-mobile-working-group/>.
  - [87] ONF, "OpenFlow Enabled Mobile and Wireless Networks, ONF Solution Brief. 2013.," 2013.
  - [88] I. F. Akyildiz, P. Wang, and S. C. Lin, "SoftAir: A software defined networking architecture for 5G wireless systems," *Comput. Networks*, 2015.
  - [89] ETSI, "ETSI GS NFV-IFA 009 V1.1.1 Network Functions Virtualisation (NFV); Management and Orchestration; Report on Architectural Options," *Etsi Gs Nfv-Ifa 009 V1.1.1*, 2016.
  - [90] L.-N. Heming, Wen; Prabhat, Kumar Tiwary; Tho, *Wireless Virtualization*. 2013.
  - [91] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wirel. Commun.*, 2017.
  - [92] M. E. Morochó Cayamcela and W. Lim, "Artificial Intelligence in 5G Technology: A Survey," in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 2018.
  - [93] J. Heinonen, P. Korja, T. Partti, H. Flinck, and P. Pöyhönen, "Mobility management enhancements for 5G low latency services," in *2016 IEEE International Conference on Communications Workshops, ICC 2016*, 2016.
  - [94] X. Xu, H. Zhang, X. Dai, Y. Hou, X. Tao, and P. Zhang, "SDN based next generation Mobile Network with Service Slicing and trials," *China Commun.*, 2014.
  - [95] H. Zhang, C. Jiang, N. C. Beaulieu, X. Chu, X. Wen, and M. Tao, "Resource allocation in spectrum-sharing OFDMA femtocells with heterogeneous services," *IEEE Trans. Commun.*, 2014.
  - [96] V. Yazici, U. C. Kozat, and M. O. Sunay, "A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management," *IEEE Commun. Mag.*, 2014.
  - [97] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for Vehicular Communications," *IEEE Commun. Mag.*, 2018.
  - [98] K. Y. Yang, H., Sun Kyoungjae, "Routing Optimization with SDN, IETF DMM Working group Internet Draft," 2016. [Online]. Available: <https://tools.ietf.org/html/draft-yang-dmm-sdn-dmm-05>.
  - [99] C. Bonnet, T.-T. Nguyen, and C. Bonnet, "SDN-based distributed mobility management for 5G networks SDN-Based Distributed Mobility Management for 5G Networks," *2016 IEEE Wirel. Commun. Netw. Conf.*, 2016.
  - [100] 3GPP TS 23.502, "Procedures for the 5G System (5GS)." Release 15, v. 15.3.0, December 2018.
  - [101] B. Hesmans, G. Detal, S. Barre, R. Bauduin, and O. Bonaventure, "SMAPP: Towards smart multipath TCP-enabled applications," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015*, 2015.
  - [102] ns-3 community, "Network Simulator 3," 2018. [Online]. Available: <https://www.nsnam.org/docs/models/html/index.html>.
  - [103] Ns-3, "WAVE Model Description." [Online]. Available: <http://www.nsnam.org/docs/models/html/wave.html>.
  - [104] M. Mezzavilla, S. Dutta, M. Zhang, M. R. Akdeniz, and S. Rangan, "5G MmWave module for the

- ns-3 network simulator,” in *MSWiM 2015 - Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2015.
- [105] ns-3 community, “mmWave module for ns-3.” [Online]. Available: <https://github.com/mmezzavilla/ns3-mmwave>.
  - [106] ns-3 community, “ns-3 Simulator Modules.” [Online]. Available: <https://www.nsnam.org/documentation/>.
  - [107] Luciano Jerez Chaves, “OpenFlow v1.3 support for ns-3.” [Online]. Available: <http://www.lrc.ic.unicamp.br/ofswitch13/>.
  - [108] OFSwitch13, “nS-3 Tap Bridge.” [Online]. Available: [http://www.lrc.ic.unicamp.br/ofswitch13/doc/html/group\\_\\_tap-bridge.html](http://www.lrc.ic.unicamp.br/ofswitch13/doc/html/group__tap-bridge.html).
  - [109] M. Dawood, M., Shaffi, “S plus mobile network architecture implementation,” 2019. [Online]. Available: <https://github.com/mshary/S-plus-Mobile-Network>.
  - [110] R. R. Roy and R. R. Roy, “Fluid-Flow Mobility,” in *Handbook of Mobile Ad Hoc Networks for Mobility Models*, 2011.
  - [111] W. Daamen, C. Buisson, and S. P. Hoogendoorn, *Traffic simulation and data: Validation methods and applications*. 2014.
  - [112] “Gnuplotting script.” [Online]. Available: <http://www.gnuplotting.org/>.
  - [113] A. Jain, E. Lopez-Aguilera, and I. Demirkol, “Enhanced handover signaling through integrated MME-SDN controller solution,” in *IEEE Vehicular Technology Conference*, 2018.
  - [114] A. Jain, E. Lopez-Aguilera, and I. Demirkol, “Improved Handover Signaling for 5G Networks,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2018.
  - [115] M. Rumney, “LTE Introduction,” *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges: Second Edition*. 2013.
  - [116] M. Dawood, W. Fuhrmann, and B. V. Ghita, “Assay of multipath TCP for session continuity in distributed mobility management,” in *Proceedings of the 11th International Network Conference, INC 2016*, 2016.
  - [117] P. Dong, J. Xie, W. Tang, N. Xiong, H. Zhong, and A. V. Vasilakos, “Performance Evaluation of Multipath TCP Scheduling Algorithms,” *IEEE Access*, 2019.
  - [118] S. Barré, C. Paasch, and O. Bonaventure, “MultiPath TCP: From theory to practice,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011.
  - [119] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, “Experimental evaluation of multipath TCP schedulers,” in *CSWS 2014 - Proceedings of the ACM SIGCOMM 2014 Workshop on Capacity Sharing Workshop*, 2014.
  - [120] Y. S. Lim, Y. C. Chen, E. M. Nahum, D. Towsley, and K. W. Lee, “Cross-layer path management in multi-path transport protocol for mobile devices,” in *Proceedings - IEEE INFOCOM*, 2014.
  - [121] A. F. M. Scharf, “RFC 6897: Multipath TCP (MPTCP) Application Interface Considerations.” [Online]. Available: <https://tools.ietf.org/html/rfc6897>.
  - [122] A. Frommgen, T. Erbschäuffer, A. Buchmann, T. Zimmermann, and K. Wehrle, “ReMP TCP: Low latency multipath TCP,” in *2016 IEEE International Conference on Communications, ICC 2016*, 2016.
  - [123] et al. J. Salim, “Linux Netlink as an IP Services Protocol. RFC 3549,” 2003.
  - [124] Redis community, “Redis database service.” [Online]. Available: <https://redis.io/>.
  - [125] M. Dawood, M., Shaffi, “S Plus Path Manager Plugin.” [Online]. Available: <https://github.com/mshary/mptcpd>.
  - [126] M. Dawood, M., Shaffi, “Socket Wrapper.” [Online]. Available: [https://github.com/mshary/socket\\_wrapper](https://github.com/mshary/socket_wrapper).
  - [127] N. Montavont, A. Blanc, R. Navas, T. Kerdoncuff, and G. Castignani, “Handover triggering in IEEE 802.11 networks,” in *Proceedings of the WoWMoM 2015*.
  - [128] W. F. Elsadek and M. N. Mikhail, “SDWM: Software defined Wi-Fi mobility for smart IoT carriers,” in *Advances in Intelligent Systems and Computing*, 2019.

## Appendix

### S+ Path Management Code

#### S Plus Plugin

```
// SPDX-License-Identifier: BSD-3-Clause
/**
 * @file splus.c
 *
 * @brief MPTCP splus path manager plugin.
 *
 * Copyright (c) 2018, 2019, Dawood
 */

#include <assert.h>
#include <stddef.h> // For NULL.
#include <limits.h>

#include <stdlib.h> // For malloc.
#include <arpa/inet.h> // For inet_ntop
#include <ifaddrs.h>
#include <sys/socket.h>
#include <netdb.h>

#include <netinet/in.h>

#include <ell/plugin.h>
#include <ell/util.h> // For L_STRINGIFY needed by l_error().
#include <ell/log.h>
#include <ell/queue.h>

#ifdef HAVE_CONFIG_H
# include <mptcpd/config-private.h> // For mptcpd VERSION.
#endif

#include <mptcpd/network_monitor.h>
#include <mptcpd/path_manager.h>
#include <mptcpd/plugin.h>

#include <hiredis/hiredis.h>

/**
 * @brief Local address to interface mapping failure value.
 */
#define SPLUS_BAD_INDEX INT_MAX
```

```

#define REDIS_HOST "127.0.0.1"
#define REDIS_PORT 6379
#define REDIS_EXPIRE 3600

/**
 * @struct splus_interface_info
 *
 * @brief Network interface information.
 *
 * This plugin tracks MPTCP connection tokens on each network
 * interface. A network interface is represented by its kernel
 * assigned index value, which is based on the local address of the
 * subflow. Once the network interface corresponding to the subflow
 * local address is determined, the connection token for that subflow
 * is then associated with the network interface as a means to denote
 * that the MPTCP connection has a subflow on that network interface.
 */
struct splus_interface_info
{
    /**
     * @brief Network interface index.
     */
    int index;

    /**
     * @brief List of MPTCP connection tokens.
     */
    struct l_queue *tokens;
};

/**
 * @struct splus_new_connection_info
 *
 * @brief Package @c new_connection() plugin operation arguments.
 *
 * This is a convenience structure for the purpose of making it easy
 * to pass @c new_connection() plugin operation arguments through
 * a single variable.
 */
struct splus_new_connection_info
{
    /**
     * @brief MPTCP connection token.
     */
    mptcpd_token_t const token;

    /**
     * @brief MPTCP local connection socket.
     */

```

```

struct sockaddr const *laddr;

/**
 * @brief MPTCP remote connection socket.
 */
struct sockaddr const *raddr;

/**
 * @brief Pointer to path manager.
 */
struct mptcpd_pm *const pm;
};

// -----

static char* get_addr(struct sockaddr const *res)
{
    char *s = NULL;
    switch(res->sa_family)
    {
        case AF_INET:
        {
            struct sockaddr_in *addr_in =
                (struct sockaddr_in *)res;
            s = malloc(INET_ADDRSTRLEN);
            inet_ntop(AF_INET,
                      &(addr_in->sin_addr),
                      s,
                      INET_ADDRSTRLEN);

            break;
        }
        case AF_INET6:
        {
            struct sockaddr_in6 *addr_in6 =
                (struct sockaddr_in6 *)res;
            s = malloc(INET6_ADDRSTRLEN);
            inet_ntop(AF_INET6,
                      &(addr_in6->sin6_addr),
                      s,
                      INET6_ADDRSTRLEN);

            break;
        }
        default:
            break;
    }
    return s;
}

/**

```

```

* @brief Match a @c sockaddr object.
*
* A network address represented by @a a (@c struct @c sockaddr)
* matches if its @c family and @c addr members match those in the
* @a b.
*
* @param[in] a Currently monitored network address of type @c struct
*              @c sockaddr*.
* @param[in] b Network address of type @c struct @c sockaddr*
*              to be compared against network address @a a.
*
* @return @c true if the network address represented by @a a matches
*         the address @a b, and @c false otherwise.
*
* @see l_queue_find()
* @see l_queue_remove_if()
*/
static bool splus_sockaddr_match(void const *a,
                                void const *b)
{
    struct sockaddr const *const lhs = a;
    struct sockaddr const *const rhs = b;

    assert(lhs);
    assert(rhs);
    assert(lhs->sa_family == AF_INET ||
           lhs->sa_family == AF_INET6);

    bool matched = (lhs->sa_family == rhs->sa_family);

    if (!matched)
    {
        return matched;
    }

    if (lhs->sa_family == AF_INET)
    {
        struct sockaddr_in const *const l =
            (struct sockaddr_in const *) lhs;
        struct sockaddr_in const *const r =
            (struct sockaddr_in const *) rhs;

        matched = (l->sin_addr.s_addr == r->sin_addr.s_addr);
    } else {
        struct sockaddr_in6 const *const l =
            (struct sockaddr_in6 const *) lhs;
        struct sockaddr_in6 const *const r =
            (struct sockaddr_in6 const *) rhs;
    }
}

```



```

        matched = (memcmp(&l->sin6_addr,
                           &r->sin6_addr,
                           sizeof(l->sin6_addr))
                   == 0);
    }

    return matched;
}

// -----

/**
 * @struct splus_nm_callback_data
 *
 * @brief Type used to return index associated with local address.
 *
 * @see @c mptcpd_nm_callback
 */
struct splus_nm_callback_data
{
    /**
     * @brief Local address information.      (IN)
     */
    struct sockaddr const* const addr;

    /**
     * @brief Network interface (link) index.  (OUT)
     */
    int index;
};

// -----

/**
 * @brief Inform kernel of local address available for subflows.
 *
 * @param[in] i      Network interface information.
 * @param[in] data   User supplied data, the path manager in this case.
 */
static void splus_send_addr(void *data,
                             void *user_data)
{
    struct sockaddr          const *const addr = data;
    struct splus_new_connection_info const *const info = user_data;

    /**
     * @bug Use real values instead of these placeholders!  The
     *       @c port, in particular, is problematic because no

```

```

    *      subflows exist for the addr in question, meaning there
    *      is no port associated with it.
    */
mptcpd_aid_t address_id = 0;

/**
 * @note The port is an optional field of the MPTCP
 *       @c ADD_ADDR option. Setting it to zero causes it to
 *       be ignored when sending the address information to
 *       the kernel.
 */
in_port_t const port = 0;

if (addr->sa_family == AF_INET)
{
    ((struct sockaddr_in*) addr)->sin_port = port;
} else {
    ((struct sockaddr_in6*) addr)->sin6_port = port;
}

if (!splus_sockaddr_match(addr, info->laddr) &&
    (addr->sa_family == info->raddr->sa_family))
{
    l_info("Broadcasting to %s", get_addr(addr));
    mptcpd_pm_send_addr(info->pm, info->token, address_id, addr);
};
}

/**
 * @brief Inform kernel of network interface usable local addresses.
 *
 * Send all local addresses associated with the given network
 * interface if that interface doesn't already have the initial
 * subflow on it.
 *
 * @param[in] i      Network interface information.
 * @param[in] data User supplied data, the path manager in this case.
 */
static void splus_send_addrs(struct mptcpd_interface const *i,
                             void *data)
{
    l_debug("interface\n"
           "  family: %d\n"
           "  type:   %d\n"
           "  index:  %d\n"
           "  flags:  0x%08x\n"
           "  name:   %s",
           i->family,
           i->type,
           i->index,
           i->flags,
           i->name,

```

```

        i->index,
        i->flags,
        i->name);

    struct splus_new_connection_info *const info = data;

    /*
     * Send each address associate with the network
     * interface.
     */
    l_queue_foreach(i->addrs, splus_send_addr, info);
}

static void splus_broadcast(mptcpd_token_t token,
                           struct sockaddr const *laddr,
                           struct sockaddr const *raddr,
                           struct mptcpd_pm *pm)
{
    /**
     * @note The kernel always provides non-zero MPTCP connection
     *       tokens.
     */
    assert(token != 0);

    struct mptcpd_nm const *const nm = mptcpd_pm_get_nm(pm);

    /*
     * Inform the kernel of additional local addresses available
     * for subflows, e.g. for MP_JOIN purposes.
     */
    struct splus_new_connection_info connection_info =
    {
        .token = token,
        .laddr = laddr,
        .raddr = raddr,
        .pm     = pm
    };

    mptcpd_nm_foreach_interface(nm,
                                splus_send_addrs,
                                &connection_info);
}

// -----
//                               Mptcpd Plugin Operations
// -----

static void splus_new_connection(mptcpd_token_t token,
                                struct sockaddr const *laddr,

```

```

        struct sockaddr const *raddr,
        struct mptcpd_pm *pm)
{
    (void) raddr;

    assert(token != 0);

    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    redisContext *c;
    redisReply *reply;

    splus_broadcast(token, laddr, raddr, pm);

    l_info("%s: new connection from %s to %s ", __func__,
           get_addr(laddr),
           get_addr(raddr));

    c = redisConnectWithTimeout(REDIS_HOST, REDIS_PORT, timeout);

    if (c != NULL && !c->err)
    {
        reply = redisCommand(c,
                             "HMSET mptcp-token-%d laddr %s raddr %s",
                             token,
                             get_addr(laddr),
                             get_addr(raddr));

        l_info("redis HMSET: %s\n", reply->str);
        freeReplyObject(reply);

        reply = redisCommand(c,
                             "EXPIRE mptcp-token-%d %d",
                             token,
                             REDIS_EXPIRE);
        freeReplyObject(reply);
    }

    redisFree(c);
}

static void splus_connection_established(mptcpd_token_t token,
        struct sockaddr const *laddr,
        struct sockaddr const *raddr,
        struct mptcpd_pm *pm)
{
    (void) raddr;

    assert(token != 0);

```

```

    if (0)          // do I need to broadcast this?
        splus_broadcast(token, laddr, raddr, pm);

    l_info("%s: connection established from %s to %s ", __func__,
           get_addr(laddr),
           get_addr(raddr));
}

static void splus_connection_closed(mptcpd_token_t token,
                                   struct mptcpd_pm *pm)
{
    (void) token;
    (void) pm;

    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    redisContext *c;
    redisReply *reply;

    c = redisConnectWithTimeout(REDIS_HOST, REDIS_PORT, timeout);

    if (c != NULL && !c->err)
    {
        reply = redisCommand(c,
                             "DEL mptcp-token-%d",
                             token);
        freeReplyObject(reply);
    }

    redisFree(c);
}

static void splus_new_address(mptcpd_token_t token,
                             mptcpd_aid_t id,
                             struct sockaddr const *addr,
                             struct mptcpd_pm *pm)
{
    (void) token;
    (void) id;
    (void) pm;

    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    redisContext *c;
    redisReply *reply;

    mptcpd_pm_send_addr(pm, token, 0, addr);

    c = redisConnectWithTimeout(REDIS_HOST, REDIS_PORT, timeout);

    if (c != NULL && !c->err)

```

```

    {
        reply = redisCommand(c,
                            "SADD mptcp-addresses %s",
                            get_addr(addr));
        freeReplyObject(reply);
    }

    redisFree(c);
}

static void splus_address_removed(mptcpd_token_t token,
                                mptcpd_aid_t id,
                                struct mptcpd_pm *pm)
{
    (void) id;
    (void) pm;

    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
    redisContext *c;
    redisReply *reply;
    char *laddr;

    c = redisConnectWithTimeout(REDIS_HOST, REDIS_PORT, timeout);

    if (c != NULL && !c->err)
    {
        reply = redisCommand(c,
                            "HGET mptcp-token-%d laddr",
                            token);
        laddr = reply->str;
        freeReplyObject(reply);

        reply = redisCommand(c, "SREM mptcp-addresses %s", laddr);
        freeReplyObject(reply);
    }

    redisFree(c);
}

static void splus_new_subflow(mptcpd_token_t token,
                              struct sockaddr const *laddr,
                              struct sockaddr const *raddr,
                              bool backup,
                              struct mptcpd_pm *pm)
{
    (void) token;
    (void) backup;
    (void) pm;
}

```

```

        l_info("%s: new subflow from %s to %s ", __func__,
               get_addr(laddr),
               get_addr(raddr));
    }

static void splus_subflow_closed(mptcpd_token_t token,
                                struct sockaddr const *laddr,
                                struct sockaddr const *raddr,
                                bool backup,
                                struct mptcpd_pm *pm)
{
    (void) token;
    (void) backup;
    (void) pm;

    l_info("%s: subflow closed from %s to %s ", __func__,
           get_addr(laddr),
           get_addr(raddr));
}

static void splus_subflow_priority(mptcpd_token_t token,
                                   struct sockaddr const *laddr,
                                   struct sockaddr const *raddr,
                                   bool backup,
                                   struct mptcpd_pm *pm)
{
    (void) token;
    (void) laddr;
    (void) raddr;
    (void) backup;
    (void) pm;

    /*
     * The splus plugin doesn't do anything with changes in subflow
     * priority.
     */
    l_warn("%s is unimplemented.", __func__);
}

static struct mptcpd_plugin_ops const pm_ops =
{
    .new_connection          = splus_new_connection,
    .connection_established = splus_connection_established,
    .connection_closed       = splus_connection_closed,
    .new_address             = splus_new_address,
    .address_removed         = splus_address_removed,
    .new_subflow             = splus_new_subflow,
    .subflow_closed          = splus_subflow_closed,
    .subflow_priority        = splus_subflow_priority
}

```

```

};

static int splus_init(void)
{
    static char const name[] = "splus";

    struct ifaddrs *addrs,*tmp;
    char host[NI_MAXHOST];
    int s;

    if (!mptcpd_plugin_register_ops(name, &pm_ops)) {
        l_error("%s: Failed to initialize splus "
                "path manager plugin.", __func__);
        return -1;
    }

    getifaddrs(&addrs);
    tmp = addrs;

    while (tmp) {
        if (tmp->ifa_addr &&
            ((tmp->ifa_addr->sa_family == AF_INET) ||
             (tmp->ifa_addr->sa_family == AF_INET6)))
        {
            s = getnameinfo(tmp->ifa_addr,
                            (tmp->ifa_addr->sa_family == AF_INET) ?
                            sizeof(struct sockaddr_in) :
                            sizeof(struct sockaddr_in6),
                            host, NI_MAXHOST,
                            NULL, 0, NI_NUMERICHOST);
            if (s == 0)
            {
                l_info("%s: Found interface %s "
                        "with IP address %s",
                        __func__,
                        tmp->ifa_name,
                        host);
            }
        }

        tmp = tmp->ifa_next;
    }

    freeifaddrs(addrs);

    l_info("MPTCP splus path manager initialized.");
    return 0;
}

```



```

static void splus_exit(void)
{
    l_info("MPTCP splus path manager exited.");
}

L_PLUGIN_DEFINE(MPTCPD_PLUGIN_DESC,
                splus,
                "SPlus path manager",
                VERSION,
                L_PLUGIN_PRIORITY_DEFAULT,
                splus_init,
                splus_exit)

/*
    Local Variables:
    c-file-style: "linux"
    End:
*/

```

## Socket Raper

```

/**
 * @file socket_wrapper.h
 *
 * @brief Override socket operations of any program
 *
 * Copyright (c) 2018, 2019, Dawood
 */
#ifndef _SOCKET_WRAPPER_
#define _SOCKET_WRAPPER_

#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <stddef.h>    // For NULL.

#include <dlfcn.h>
#include <errno.h>
#include <linux/tcp.h>

#include <string.h>

/**
 * @brief original socket method
 */
static int (*__socket)(int domain, int type, int protocol) = NULL;

/**
 * @brief original setsockopt method

```

```

*/
static int (*__setsockopt)(int sockfd, int level, int optname, const void
*optval, socklen_t optlen) = NULL;

/**
 * @brief original bind method
 */
static int (*__bind)(int sockfd, const struct sockaddr *addr, socklen_t len)
= NULL;

/**
 * @brief original listen method
 */
static int (*__listen)(int sockfd, int backlog) = NULL;

/**
 * @brief original accept method
 */
static int (*__accept)(int sockfd, struct sockaddr *addr, socklen_t *len) =
NULL;

/**
 * @brief original connect method
 */
static int (*__connect)(int sockfd, const struct sockaddr *addr, socklen_t
len) = NULL;

/**
 * @brief original close method
 */
static int (*__close)(int sockfd) = NULL;

/**
 * @brief original read method
 */
static ssize_t (*__read)(int sockfd, void *buf, size_t c) = NULL;

/**
 * @brief original write method
 */
static ssize_t (*__write)(int sockfd, void *buf, size_t c) = NULL;

/**
 * @brief override method - socket
 */
int socket(int domain, int type, int protocol);

/**
 * @brief override method - setsockopt

```

```

*/
int setsockopt(int sockfd, int level, int optname, const void *optval,
socklen_t optlen);

/**
 * @brief override method - bind
 */
int bind(int sockfd, const struct sockaddr *addr, socklen_t len);

/**
 * @brief override method - listen
 */
int listen(int sockfd, int backlog);

/**
 * @brief override method - accept
 */
int accept(int sockfd, struct sockaddr *addr, socklen_t *len);

/**
 * @brief override method - connect
 */
int connect(int sockfd, const struct sockaddr *addr, socklen_t len);

/**
 * @brief override method - close
 */
int close(int sockfd);

/**
 * @brief override method - read
 */
ssize_t read(int sockfd, void *buf, size_t c);

/**
 * @brief override method - write
 */
ssize_t write(int sockfd, void *buf, size_t c);

#endif

Mptcp_wrapper.h

/**
 * @file mptcp_wrapper.h
 *
 * @brief MPTCP configuration setup on per socket basis
 *
 * Copyright (c) 2018, 2019, Dawood

```

```

*/
#ifndef _MPTCP_WRAPPER_
#define _MPTCP_WRAPPER_

#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <linux/tcp.h>
#include <string.h>

#define MPTCP_INFO_FLAG_SAVE_MASTER 0x01

/**
 * @brief gets the MPTCP information from an existing socket
 *
 * @param[in] fd      socket handle
 * @param[in] level  socket type e.g. SOL_TCP
 * @param[out] ret    the @c struct @mptcp_info to stored MPTCP info
 */
static int get_mptcp_info(int fd, int level, struct mptcp_info *ret);

/**
 * @brief sets optimal MPTCP configuration on an existing socket
 *
 * @param[in] sockfd socket handle
 * @param[in] level  socket type e.g. SOL_TCP
 * @return      value from original setsockopt method
 */
static int set_mptcp_options(int sockfd, int level);

#endif

mptcp_wrapper.c

/**
 * @file mptcp_wrapper.c
 *
 * @brief MPTCP configuration setup on per socket basis
 *
 * Copyright (c) 2018, 2019, Dawood
 */
#include "mptcp_wrapper.h"
#include "socket_wrapper.h"

/**
 * @brief gets the MPTCP information from an existing socket
 *
 * @param[in] fd      socket handle
 * @param[in] level  socket type e.g. SOL_TCP

```

```

* @param[out] ret    the @c struct @mptcp_info to stored MPTCP info
*/
static int get_mptcp_info(int fd, int level, struct mptcp_info *ret) {
    struct mptcp_info minfo;
    struct mptcp_meta_info meta_info;
    struct tcp_info initial;
    struct tcp_info others[3];
    struct mptcp_sub_info others_info[3];

    minfo.tcp_info_len = sizeof(struct tcp_info);
    minfo.sub_len = sizeof(others);
    minfo.meta_len = sizeof(struct mptcp_meta_info);
    minfo.meta_info = &meta_info;
    minfo.initial = &initial;
    minfo.subflows = others;
    minfo.sub_info_len = sizeof(struct mptcp_sub_info);
    minfo.total_sub_info_len = sizeof(others_info);
    minfo.subflow_info = others_info;
    ret = &minfo;

    return getsockopt(fd, level, MPTCP_INFO, &minfo, (socklen_t
*)sizeof(minfo));
}

/**
 * @brief sets optimal MPTCP configuration on an existing socket
 *
 * @param[in] sockfd socket handle
 * @param[in] level  socket type e.g. SOL_TCP
 * @return      value from original setsockopt method
 */
static int set_mptcp_options(int sockfd, int level) {
    if (sockfd != 0 && level == IPPROTO_TCP) {
        int enable = 1;
        int ret = __setsockopt(sockfd, level, MPTCP_ENABLED, &enable,
sizeof(enable));

        if (ret < 0) {
            fprintf(stderr, "setsockopt: MPTCP_ENABLED error %s!\n",
strerror(errno));
            fflush(stderr);
            return ret;
        }

        char pathmanager[] = "fullmesh";
        ret = __setsockopt(sockfd, level, MPTCP_PATH_MANAGER,
pathmanager, sizeof(pathmanager));

        if (ret < 0) {

```

```

        fprintf(stderr, "setsockopt: MPTCP_PATH_MANAGER error
%s!\n", strerror(errno));
        fflush(stderr);
        return ret;
    }

    char scheduler[] = "default";
    ret = __setsockopt(sockfd, level, MPTCP_SCHEDULER, scheduler,
sizeof(scheduler));

    if (ret < 0) {
        fprintf(stderr, "setsockopt: MPTCP_SCHEDULER error %s!\n",
strerror(errno));
        fflush(stderr);
        return ret;
    }

    int val = MPTCP_INFO_FLAG_SAVE_MASTER;
    ret = __setsockopt(sockfd, level, MPTCP_INFO, &val, sizeof(val));

    if (ret < 0) {
        fprintf(stderr, "setsockopt: MPTCP_INFO error %s!\n",
strerror(errno));
        fflush(stderr);
    }

    return ret;
}

return 0;
}

```

Socket\_wrapper.c

```

/**
 * @file socket_wrapper.c
 *
 * @brief Override socket operations of any program
 *
 * To load and apply this wrapper to any program,
 * LD_PRELOAD=path_to_wrapper_library/libsocket_wrapper.so ./your_program
 *
 * e.g.
 * LD_PRELOAD=/root/Work/libsocket_wrapper.so ssh root@sip-server.de
 *
 * Copyright (c) 2018, 2019, Dawood
 */

#include "socket_wrapper.h"

```

```

#ifdef WITH_REDIS
#include <hiredis/hiredis.h>

#define REDIS_HOST "/var/run/redis/redis-server.sock"
#endif

static int run_once = 0;

/**
 * @brief override method - socket
 */
int socket(int domain, int type, int protocol) {
    char *msg;

    if (__socket == NULL) {
        __socket = dlsym(RTLD_NEXT, "socket");
    }

#ifdef DEBUG
    fprintf(stderr, "Loading __socket => %p\n", __socket);
    fflush(stderr);
#endif

    if ((msg = dlerror()) != NULL) {
        fprintf(stderr, "socket: dlopen failed: %s\n", msg);
        fflush(stderr);
        exit(1);
    }

    return __socket(domain, type, protocol);
}

/**
 * @brief override method - setsockopt
 */
int setsockopt(int sockfd, int level, int optname, const void *optval,
socklen_t optlen) {
    char *msg;

#ifdef WITH_REDIS
    int no_delay = -1;
    redisContext *c;
    redisReply *reply;
    struct timeval timeout = { 1, 500000 }; // 1.5 seconds
#endif

    if (__setsockopt == NULL) {
        __setsockopt = dlsym(RTLD_NEXT, "setsockopt");
    }

```

```

#ifdef DEBUG
    fprintf(stderr, "Loading __setsockopt => %p\n", __setsockopt);
    fflush(stderr);
#endif

    if ((msg = dlerror()) != NULL) {
        fprintf(stderr, "setsockopt: dlopen failed: %s\n", msg);
        fflush(stderr);
        exit(1);
    }
}

#ifdef WITH_REDIS
    // load tcp no-delay option from redis and apply it only once
    if (level == IPPROTO_TCP && run_once == 0) {
        c = redisConnectUnixWithTimeout(REDIS_HOST, timeout);
        if (c != NULL && !c->err) {
            reply = redisCommand(c, "GET tcp-no-delay");
            if (c != NULL && !c->err) {
                no_delay = reply->integer;
            }
            freeReplyObject(reply);
            redisFree(c);
        }
        if (no_delay >= 0) {
            fprintf(stderr, "setsockopt:  TCP_NODELAY  =>  %d\n",
no_delay);

            fflush(stderr);

            __setsockopt(sockfd, IPPROTO_TCP, TCP_NODELAY, &no_delay,
sizeof(no_delay));
        }
        run_once = 1;
    }
#endif

    return __setsockopt(sockfd, level, optname, optval, optlen);
}

/**
 * @brief override method - bind
 */
int bind(int sockfd, const struct sockaddr *addr, socklen_t len) {
    char *msg;

    if (__bind == NULL) {
        __bind = dlsym(RTLD_NEXT, "bind");
    }

```



```

#ifdef DEBUG
    fprintf(stderr, "Loading __bind => %p\n", __bind);
    fflush(stderr);
#endif

    if ((msg = dlerror()) != NULL) {
        fprintf(stderr, "bind: dlopen failed: %s\n", msg);
        fflush(stderr);
        exit(1);
    }
}

return __bind(sockfd, addr, len);
}

/**
 * @brief override method - listen
 */
int listen(int sockfd, int backlog) {
    char *msg;

    if (__listen == NULL) {
        __listen = dlsym(RTLD_NEXT, "listen");
    }

#ifdef DEBUG
    fprintf(stderr, "Loading __listen => %p\n", __listen);
    fflush(stderr);
#endif

    if ((msg = dlerror()) != NULL) {
        fprintf(stderr, "listen: dlopen failed: %s\n", msg);
        fflush(stderr);
        exit(1);
    }
}

return __listen(sockfd, backlog);
}

/**
 * @brief override method - accept
 */
int accept(int sockfd, struct sockaddr *addr, socklen_t *len) {
    char *msg;

    if (__accept == NULL) {
        __accept = dlsym(RTLD_NEXT, "accept");
    }

#ifdef DEBUG

```

```

        fprintf(stderr, "Loading __accept => %p\n", __accept);
        fflush(stderr);
#endif

        if ((msg = dlerror()) != NULL) {
            fprintf(stderr, "accept: dlopen failed: %s\n", msg);
            fflush(stderr);
            exit(1);
        }
    }

    return __accept(sockfd, addr, len);
}

/**
 * @brief override method - connect
 */
int connect(int sockfd, const struct sockaddr *addr, socklen_t len) {
    char *msg;

    if (__connect == NULL) {
        __connect = dlsym(RTLD_NEXT, "connect");
    }

#ifdef DEBUG
        fprintf(stderr, "Loading __connect => %p\n", __connect);
        fflush(stderr);
#endif

    if ((msg = dlerror()) != NULL) {
        fprintf(stderr, "connect: dlopen failed: %s\n", msg);
        fflush(stderr);
        exit(1);
    }

    return __connect(sockfd, addr, len);
}

/**
 * @brief override method - close
 */
int close(int sockfd) {
    char *msg;

    if (__close == NULL) {
        __close = dlsym(RTLD_NEXT, "close");
    }

#ifdef DEBUG
        fprintf(stderr, "Loading __close => %p\n", __close);

```

```

        fflush(stderr);
#endif

        if ((msg = dlerror()) != NULL) {
            fprintf(stderr, "close: dlopen failed: %s\n", msg);
            fflush(stderr);
            exit(1);
        }
    }

    return __close(sockfd);
}

/**
 * @brief override method - read
 */
ssize_t read(int sockfd, void *buf, size_t c) {
    char *msg;

    if (__read == NULL) {
        __read = dlsym(RTLD_NEXT, "read");
    }

#ifdef DEBUG
    fprintf(stderr, "Loading __read => %p\n", __read);
    fflush(stderr);
#endif

    if ((msg = dlerror()) != NULL) {
        fprintf(stderr, "read: dlopen failed: %s\n", msg);
        fflush(stderr);
        exit(1);
    }
}

    return __read(sockfd, buf, c);
}

ssize_t write(int sockfd, void *buf, size_t c) {
    char *msg;

    if (__write == NULL) {
        __write = dlsym(RTLD_NEXT, "write");
    }

#ifdef DEBUG
    fprintf(stderr, "Loading __write => %p\n", __write);
    fflush(stderr);
#endif

    if ((msg = dlerror()) != NULL) {

```

```

        fprintf(stderr, "write: dlopen failed: %s\n", msg);
        fflush(stderr);
        exit(1);
    }
}

return __write(sockfd, buf, c);
}

```

## Gateway manager python script

```

#!/usr/bin/python3
#
# This demo script use Linux NETLINK IPC to monitor
# and manage default network gateway.
#
# It reads REDIS for preferred network interface device
# and IPv4 address. Whenever that device comes online
# with that IPv4 address, it changes the default network
# gateway to that device.
#
# A typical use case of this script is mobility management
# of MPTCP enabled UE in 5G service based architecture (SBA).
# The script switches to fastest network connection
# (configureable via REDIS backend service) as soon as it
# becomes available without breaking any active communication
# session.
#
# Copyright (c) 2018, 2019, Dawood
#
import redis
from socket import AF_INET
from pyroute2 import IPRoute
from pprint import pprint

# debug settings
debug = 0

# redis settings
redis_host = 'localhost'
redis_port = 6379
redis_db = 0

# default settings - used when redis is unavailable
default_inet = 'wlx801f02cd4b98'
default_ipv4 = '192.168.8.140'

```

```
#####

def get_preferences():
    r = redis.Redis(host=redis_host, port=redis_port, db=redis_db)
    ret = dict()

    preferred_inet = r.get('GET pref-inet')
    if not preferred_inet:
        preferred_inet = default_inet
    print("Preferred Interface:", preferred_inet)
    ret['inet'] = preferred_inet

    preferred_ipv4 = r.get('GET pref-ipv4')
    if not preferred_ipv4:
        preferred_ipv4 = default_ipv4
    print("Preferred IPv4:", preferred_ipv4)
    ret['ipv4'] = preferred_ipv4

    with IPRoute() as ipr:
        for x in ipr.get_routes(table=254):
            dev = ipr.link_lookup(ifname=preferred_inet)
            if (dev and x.get_attr('RTA_OIF') == dev[0]):
                route = dict()
                route['dev'] = x.get_attr('RTA_OIF')
                route['gw'] = x.get_attr('RTA_GATEWAY')
                route['prio'] = x.get_attr('RTA_PRIORITY')
                print("Preferred Route:", route, "\n")
                ret['route'] = route
                break

        check = False
        for x in ipr.get_addr(family=AF_INET, label=preferred_inet):
            if x.get_attr('IFA_ADDRESS') == preferred_ipv4:
                check = True
                break

        if (check):
            if debug:
                print("Set preferences:", preferred_ipv4, "assigned to",
                    preferred_inet, "and route", route, "\n")
        else:
            if debug:
                print("Bad preferences:", preferred_ipv4,
                    "is not assigned to", preferred_inet)

    return ret

prefs = get_preferences()
```

```

print("Current Network Status:", prefs, "\n")

# With IPRoute objects you have to call bind() manually
with IPRoute() as ipr:
    ipr.bind()
    while 1:
        for message in ipr.get():
            # print all messages - for debugging only
            #pprint(message)

            # process only new IPv4 address events
            if (message.get('event') == 'RTM_NEWADDR' and
                message.get('family') == AF_INET):
                # print all new ip messages - for debugging only
                #pprint(message)
                #print("\n")

                prefs = get_preferences()
                if (prefs['route']['dev'] == message.get('index') and
                    prefs['route']['gw']):
                    if debug:
                        pprint(message)
                        print("\n")

                print("Changing default gateway to",
                      prefs['route']['gw'])

                # delete default gateway if exists
                ipr.route("del", dst="default")

                # add new default gateway
                ipr.route("add", dst="default",
                          gateway=prefs['route']['gw'])

```